PROJECTS ■ THEORY ■ APPLICATIONS ■ CIRCUITS ■ TECHNOLOGY

NUTS AND VOLTS

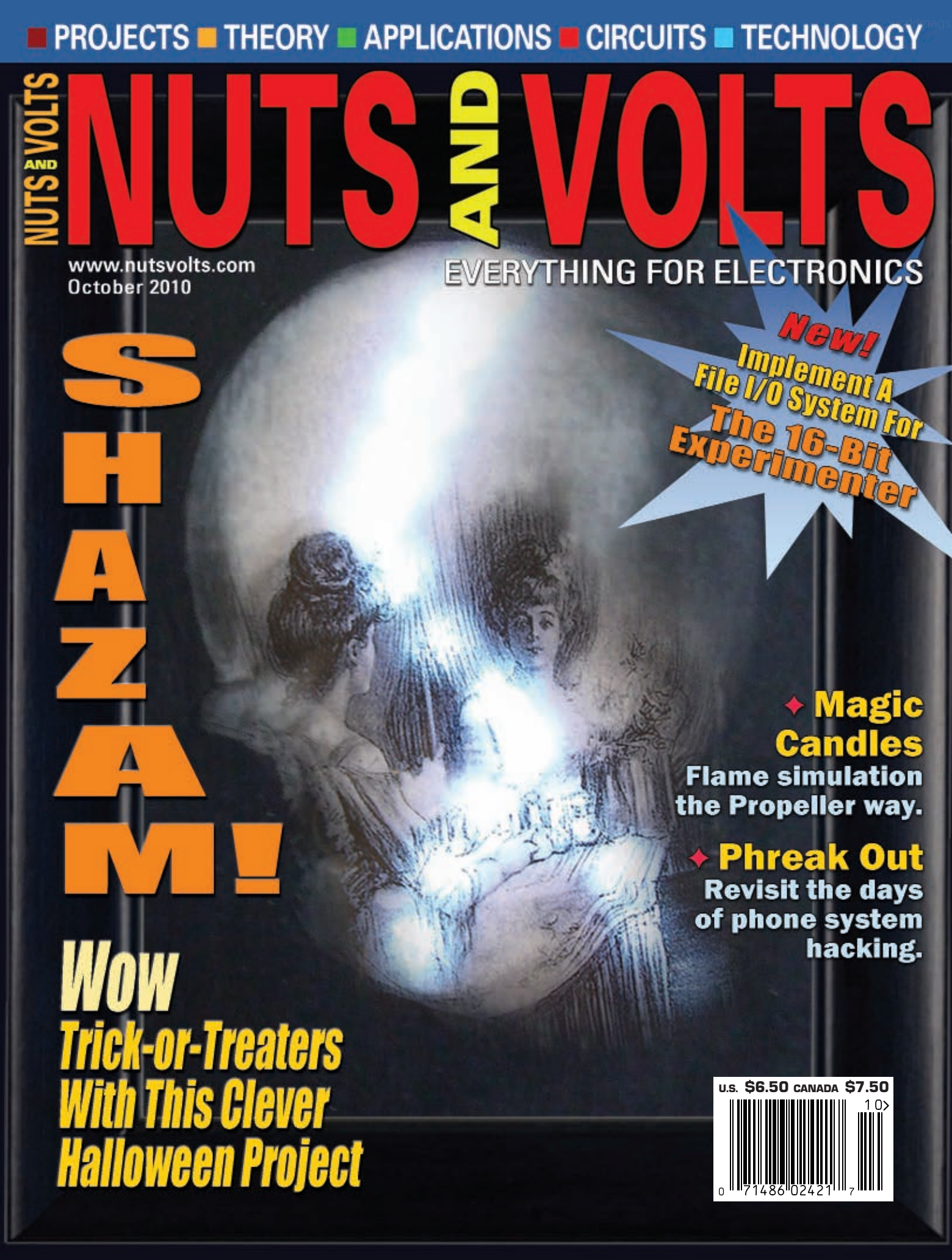
NUTS AND VOLTS

www.nutsvolts.com
October 2010

EVERYTHING FOR ELECTRONICS

SHAZAM!

New!
Implement A File I/O System For The 16-Bit Experimenter



Magic Candles
Flame simulation the Propeller way.

Phreak Out
Revisit the days of phone system hacking.

Wow Trick-or-Treaters With This Clever Halloween Project

U.S. $6.50 CANADA $7.50
10
0 71486 02421 7

All displays shown are actual size

SerialLCD™

BasicX.com

NetMedia

RoHS 2002/95/EC COMPLIANT

NetMedia Inc., 10940 N Stallard Pl, Tucson Arizona 85737 tel.520.544.4567 fax.520.544.0800 www.netmedia.com

worldmags

Power Your Connected Graphics Solution With the PIC32 Microcontroller

Connectivity and graphical user interfaces are essential in today's applications. You're challenged to deliver intuitive, high impact, connected solutions while maintaining flexibility to support several different product options. Microchip's PIC32 series of 32-bit microcontrollers offer the right performance, memory size and peripherals to help achieve your goals.

With 1.56 DMIPS/MHz performance topping any device in its category, up to 512 Kbytes of Flash, 128 Kbytes of RAM and integrated connectivity peripherals like Ethernet, CAN and USB the PIC32 can deliver the mix of performance and flexibility needed to help you meet your design challenges.

Microchip gets you there with:

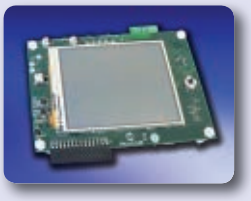
- PIC32 Starter Kits – Standalone easy to use development boards with integrated debugger/programmer
- Multimedia Expansion Board – The most complete user interface development solution in it's class – enabling development of highly interactive, graphics and audio-based interfaces with WiFi connectivity – a modular add-on to any PIC32 Starter Kit.
- Microchip's FREE Graphics and Connectivity libraries and code examples – Eases your development effort and speeds your time to market

GET STARTED IN 3 EASY STEPS

1. Purchase PIC32 Ethernet Starter Kit and Multimedia Expansion Board
2. Download MPLAB® IDE
3. Start designing!

www.microchip.com/graphics

PIC32 Ethernet Starter Kit – DM320004



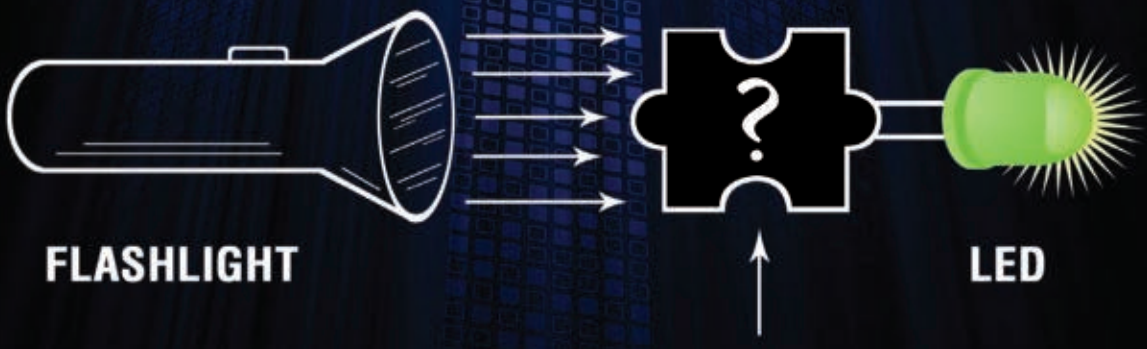
Multimedia Expansion Board - DM320005

The Microchip name and logo, the Microchip logo, PIC and MPLAB are registered trademarks and PICtail is a trademark of Microchip Technology Incorporated in the USA and in other countries. All other trademarks mentioned herein are the property of their respective owners.
© 2010, Microchip Technology Incorporated. All rights reserved. 9/10

MICROCHIP

worldmags

Can You Solve This Puzzle?



What is the missing component?

JAMECO
ELECTRONICS
REACH FOR JAMECO FIRST | FOR ALL YOUR ELECTRONIC NEEDS
GREG'S CORNER
DO IT YOURSELF
Jameco Summer School
Make:
Learn how to use a perfboard with this step-by-step video from Maker-extraordinaire Collin Cunningham. Time to fire up that soldering iron and learn how to build your own prototype.
Get your 8 minute education here...
WHAT'S IN THE NEWS
Wireless Sensor, Actuator Networks and You

A Physics professor planned to illustrate a lecture with demonstrations of how light can be intercepted by certain phosphors or various optoelectronic sensors and transformed into entirely new light. He wanted an ultra-simple demonstration of how an LED would glow when biased by a forward current provided by a suitable sensor. He rummaged through his optoelectronic drawer and found two silicon solar cells, several cadmium sulfide photo resistors, a couple of AlGaAs red LEDs, some silicon phototransistors and half a dozen silicon photodiodes. In his parts cabinet he found some transistors, miniature chokes and assorted resistors and capacitors. How did he combine the smallest number of components to do what the phosphor card did?
Go to www.Jameco.com/search8 to see if you are correct.

Sign up for our e-newsletter and get a bonus puzzle.
www.Jameco.com/enews

JAMECO® ELECTRONICS

1-800-831-4242 | www.Jameco.com

worldmags

LeCroy 40 MHz - 1 GHz Oscilloscopes

Performance you need at a price you can afford

Starting at $695

WaveSurfer®
200 MHz - 1 GHz

WaveJet®
100 MHz - 500 MHz

WaveAce™
40 MHz - 300 MHz

Save 15% when you trade in your old scope!

Contact a distributor today

LeCroy Authorized Distributors:

AVALON Equipment Corporation
Avalon Equipment Corp.
(888)-542-8256
avalontest.com

TREK EQUIPMENT
Trek Equipment
(888) 873-5374
trekequipment.com

STANLEY Supply & Services
Stanley Supply & Services
(800) 225-5370
StanleySupplyServices.com

Tequipment.NET
Tequipment.NET
(877) 742-8378
tequipment.net

KEMP
Kemp Instruments
(888) 619-2266
kempinstruments.com

MEUNIER
Meunier Elect. Supply
(800) 638-6437
meunierusa.com

GLOBAL TEST SUPPLY
The Test, Measurement & Supply Co!
Global Test Supply
888-610-7664
globaltestsupply.com

LeCroy

To learn more about LeCroy's full line of oscilloscopes, protocol analyzers and embedded test tools visit www.lecroy.com or call 1-800-5-LeCroy

worldmags

worldmags

Nuts & Volts October 2010

www.nutsvolts.com

Did you know that each article in *Nuts & Volts* has its own webpage? It's where you go for downloads, comments, updates, corrections, or to link to the article in the digital issue. The unique link for each webpage is included at the top of the article.

You can also visit article pages from back issues at **www.nutsvolts.com**. Just select the ***Back Issues*** tab from the ***Magazine*** drop down menu, click the Table of Contents link, and the article name.

Projects & Features

32 Build the Shazam!!
Light up the eyes of trick-or-treaters this year with lightning bolts and thunder claps which are hidden within a photo frame.
■ *By Ron Newton*

38 Make Magic Candles with the Propeller
Discover a cool way to safely simulate flames for all your holiday displays.
■ *By Jon McPhalen*

46 Implementing a File I/O System for the 16-bit Micro Experimenter
Add even more capability to your Experimenter by integrating a PC compatible file I/O.
■ *By Thomas Kibalo*

52 Phreak Out with the BASIC Stamp 2
Remember the old days of hacking/phreaking with your cell phone? Well, you can revisit those times (or try them out for the first time) in a fun, legally safe environment.
■ *By Abraham Smith*

FUNdamentals For Beginners

Just getting started in electronics? Try these simple demos to illustrate electronics principles and how components function in actual circuits!

76 How an SCR Works

Columns

10 TechKnowledgey 2010
Events, Advances, and News
Wastewater that produces electricity, custom gaming computer, 2.5-D images by Christmas, plus other stuff you'll find interesting.

14 The Design Cycle
Advanced Techniques for Design Engineers
Thumbs Up for the Vinculum-II Toolchain.

24 Q & A
Reader Questions Answered Here
High voltage sine wave, obscure battery, black body heat source, plus more.

57 Open Communication
The Latest in Networking and Wireless Technologies
Q & A about Digital Radio.

60 Smiley's Workshop
Programming • Hardware • Projects
AVR Memory Part 5: Bootloaders.

67 PICAXE Primer
Sharpening Your Tools of Creativity
Implementing an ADC Keypad.

Departments

08	DEVELOPING PERSPECTIVES	72	NV WEBSTORE
44	SHOWCASE	76	CLASSIFIEDS
66	ELECTRO-NET	78	TECH FORUM
		81	AD INDEX

Nuts & Volts (ISSN 1528-9885/CDN Pub Agree #40702530) is published monthly for $26.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL MAILING OFFICES. POSTMASTER: Send address changes to **Nuts & Volts, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@nutsvolts.com.

worldmags

Low-cost Industrial
Serial to **Ethernet** Solutions

NetBurner
Networking in One Day!

- Instantly network-enable any serial device
- No programming is required for serial to Ethernet application
- Customize to suit any application with a development kit

SBL2e Chip

2-port serial-to-Ethernet server with eight A/D converter inputs and optional SPI, I2C, CAN peripheral device support

SBL2e 200

2-port serial-to-Ethernet server with four A/D converter inputs, optional I2C peripheral support and 10-pin header

SBL2e 100

2-port serial-to-Ethernet server with four A/D converter inputs, optional I2C peripheral support and RJ45 connector

SBL2e X

External 2-port serial-to-Ethernet server with RS-232 support

SBL2e XA

External server with up to four A/D converter inputs, up to eight digital I/O and up to two UARTS (one RS-232)

Hardware Features

Up to three serial ports, 10/100 Mbps Ethernet, up to 10 digital I/O, 12-bit A/D converters, operating temperature -40 to 85 C, 32-bit performance

Software Features

TCP/UDP/Telnet/HTTP modes, DHCP/Static IP support, web-based or AT command configuration

Low Prices

SBL2e Chip ..... $10.50 (Qty. 10k, Device P/N: SBL2eCHIP-236IR)
SBL2e 200 ..... $19.95 (Qty. 1000, Device P/N: SBL2e-200IR)
SBL2e 100 ..... $21.95 (Qty. 1000, Device P/N: SBL2e-100IR)
SBL2e X ..... $79.00 (Qty. 1000, Device P/N: SBL2eX-100IR)
SBL2e XA ..... $79.00 (Qty. 1000, Device P/N: SBL2eXA-100IR)

Need a custom solution?

NetBurner Serial to Ethernet Development Kits are available to customize any aspect of operation including web pages, data filtering, or custom network applications. All kits include platform hardware, ANSI C/C++ compiler, TCP/IP stack, web server, e-mail protocols, RTOS, and NB Eclipse IDE.

Information and Sales | sales@netburner.com

Web | www.netburner.com **Telephone** | 1-800-695-6828

worldmags

by Bryan Bergeron, Editor

DEVELOPING PERSPECTIVES

Electronics Enthusiast - Is It Time To Redefine The Term?

I had lunch with a fellow electronics enthusiast on his return from a trip with his daughter in search of a college of engineering. He was blown away by the level of electronic automation in many of the campuses they visited. Not only was there ubiquitous Wi-Fi, but at several colleges, smartphone-toting students could monitor the status of the washing machines and dryers in each of the dorms. Tablet-sized LCD panels affixed to classroom doors announced activities in that room for the day, and numerous PC kiosks enabled students to check their email or class schedule. My friend's daughter wasn't as impressed with the modern electronic accoutrements, but instead expected what they found and more. Perhaps this explained my friend's frustration with trying to teach his daughter the "old fashioned" electronics of resistors, transistors, and other discrete components. With sights set on creating a startup company using mobile computing, she couldn't see the relevance learning to flip a few LEDs on and off with a complex circuit or of building a power supply from scratch when a single chip would do the job.

My automatic response was that learning the fundamentals is, well, fundamental. If you're going to design or build electronics, you have to have a handle on the fundamentals. We decided that sooner or later, she would come around and delve into the basic physics and discrete components the way we did when we discovered electronics. However, I began to have second thoughts soon after the meeting. Is it really necessary to deal with capacitors, resistors, and discrete components to understand and — more importantly — leverage modern electronics? Is it a waste of time for someone planning to create a business based on new devices to play with and learn to use 'outdated' components? Is it better to move directly to, say, microcontrollers and programmable field gate arrays (FPGAs)?

As an educator, I can say that knowledge of the fundamentals is important, but you have to be selective in the fundamentals you decide to focus on. I remember in college as an engineering major that knowledge of Quantum Mechanics was deemed critical to understanding how semiconductors work. So, I suffered through the course. To this day, I can't say that I've applied any of what I might have learned to semiconductor design. On the other hand, if I had chosen to work for Intel or some other chip manufacturer, perhaps Quantum Mechanics would have served me well.

You no doubt have your own reasons for reading the articles in *Nuts & Volts*. You may be new to electronics, and every bit of information is new and exciting. Perhaps you're retired and like the familiarity of the 'old fashioned' circuitry. Maybe you're studying engineering and you're using the articles on microcontrollers as supplemental material in your studies. Perhaps you simply like the satisfaction of building circuits with your hands. Many of your fellow readers are specialists in other fields and have an idea of automating some aspect of their work and look to this magazine to provide a soft introduction to the world of applied electronics.

After you read through this issue, I'd like you to email me with your thoughts on the overall focus of the magazine. Does it define — in your view — the world of the electronics enthusiast? As the field of electronics evolves, do you want us to cover the bleeding edge, or are you more comfortable working a little behind the curve, where components are cheaper and better documented? Again, I'd like to hear from you. I'll post the results of this informal survey in an upcoming editorial. **NV**

Backpacks make great things greater.
You know, like when Luke carried Yoda around on Dagobah.
Shown with Ardweeny & TMP36 installed (not included)
SKU: KARDWBP
Price: $11.95
The Ardweeny Backpack adds full portability, 5V regulation, Servo, Temperature & Blink-M features to the Arduino-compatible Ardweeny!
SOLARBOTICS ltd
www.solarbotics.com
1-866-276-2687
PS- We were going to mention broken C-3PO on Chewbacca, but that was more of a mesh than a backpack.

worldmags

NUTS AND VOLTS
EVERYTHING FOR ELECTRONICS

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct.
Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore orders only **1-800-783-4624**
www.nutsvolts.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277
North Hollywood, CA 91615

FOUNDER/ASSOCIATE PUBLISHER
Jack Lemieux

PUBLISHER
Larry Lemieux
publisher@nutsvolts.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@nutsvolts.com

EDITOR
Bryan Bergeron
techedit-nutsvolts@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Joe Pardue, Ron Hackett, Ron Newton, Abe Smith, Russ Kincaid, Fred Eady, Lou Frenzel, Jon McPhalen, Thomas Kibalo

CIRCULATION DIRECTOR
Tracy Kerley
subscribe@nutsvolts.com

SHOW COORDINATOR
Audrey Lemieux

MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@nutsvolts.com

WEB CONTENT
Michael Kaudze
website@nutsvolts.com

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION/GRAPHICS
Shannon Christensen

Copyright © 2010 by T & L Publications, Inc.
All Rights Reserved
All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *Nuts & Volts Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *Nuts & Volts*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *Nuts & Volts*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: **430 Princeland Court, Corona, CA 92879**.

Printed in the USA on SFI & FSC stock.

NUTS AND VOLTS
Nuts & Volts 6 CD-ROMs & Hat Special!
That's 72 issues. Complete with supporting code and media files.
NUTS VOLTS 2009
NUTS VOLTS 2008
NUTS VOLTS 2007
NUTS VOLTS 2006
Nuts & Volts 2005
Nuts & Volts 2004
Free Shipping!
Only $129.95
NUTS AND VOLTS

Electronic Surplus

sales1@electronicsurplus com
Toll free: 1-800-642-1123
8755 Munson Rd
Cleveland, OH 44060

electronicsurplus.com/specials.cstm

Geared Stepper motor 5V 10:1 — **$14.95**

Lever switch 20A N.C. pkg of 4 — **$3.00**

Transformer dual-primary 8V 750mA — **$8.00**

Motor 3-12VDC 8400 RPM with tach — **$8.00**

T-1 Photo-transistor visible & NIR pkg of 4 — **$1.00**

Relay 20A N.O. 120VAC — **$4.95**

EARN MORE MONEY
Get your dream job!

Be an FCC Licensed Wireless Technician!

Make up to $100,000 a year and more with NO college degree

Learn Wireless Communications and get your "FCC Commercial License" with our proven Home-Study Course!

- No need to quit your job or go to school.
- This course is easy, fast and low cost.
- No previous experience needed!
- Learn at home in your spare time!

Move to the front of the employment line in Radio-TV, Communications, Avionics, Radar, Maritime and more… even start your own business!

Call now for FREE info

800-932-4268 ext. 209

Or, email us:
fcc@CommandProductions.com

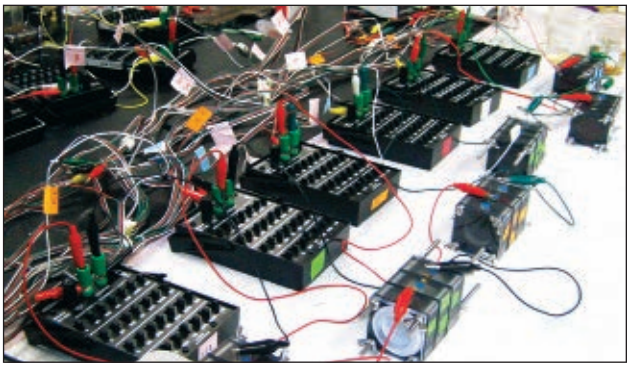
COMMAND PRODUCTIONS
Warren Weagant's FCC License Training
P.O. Box 3000, Dept. 209 • Sausalito, CA 94966
Please rush FREE info immediately!
NAME: _____
ADDRESS: _____
CITY/STATE/ZIP: _____
You may fax your request to: 415-332-1901

TECH KNOWLEDGEY 2010

EVENTS, ADVANCES, AND NEWS

■ BY JEFF ECKERT

ADVANCED TECHNOLOGY



■Three-chambered microbial desalination cells in the laboratory. *Photo courtesy of David Jones, Penn State.*

WASTEWATER PRODUCES ELECTRICITY AND DESALINATION

There is, of course, a scarcity of clean water for drinking, washing, and other critical uses in many parts of the world, so it would be pretty nice if we could accomplish desalination without relying on reverse osmosis or energy-hogging electrodialysis. It turns out that Bruce Logan, an environmental engineering professor at Penn State (**www.psu.edu**), along with a cadre of researchers from China's Tsinnghua University, have found a way — at least in theory. The team accomplished the feat with a modified microbial fuel cell which uses bacteria to simultaneously clean the water and generate electricity. A typical cell consists of two chambers — one holding the wastewater and the other just plain water — with an electrode in each one. Naturally occurring bacteria in the former consume organic materials and generate a current. The Penn version adds a third chamber in between with ion-specific membranes (i.e., membranes that allow either positive or negative ions — but not both — to pass through) between the central chamber and the electrodes. Salty water is placed in the central chamber. As the bacteria consume the water, negative ions move from the salty water to the wastewater chamber, and positive ions move to the other electrode chamber, thereby desalinating water in the central one. "When we try to use microbial fuel cells to generate electricity, the conductivity of the wastewater is very low," said Logan. "If we could add salt, it would work better. Rather than just add in salt, however, in places where brackish or salt water is already abundant, we could use the process to additionally desalinate salty water, clean the wastewater, and dump it and the resulting salt back into the ocean." Before you get too excited, though, note that there is a catch. It took 200 mL of wastewater (in this case, acetic acid in water) to desalinate 3 mL of saltwater. As Logan observed, "This is not a practical system yet, as it is not optimized, but it is proof of concept." ▲

BYE-BYE BLOODHOUNDS

It may sound a bit morbid, but the National Institute of Standards and Technology (NIST, **www.nist.gov**) recently demonstrated a new technique for locating grave sites. In the past, cadaver-sniffing dogs or ground penetrating radar has been used for such things, but neither is all that effective at finding bodies that are hidden by concrete or otherwise disguised. As described by NIST, the new process "uses an alumina-coated porous-layer open tubular (PLOT) column with a motorized pipette that pulls in air samples at ambient temperatures. The device detects trace amounts of ninhydrin-reactive nitrogen (NRN) that collects in air pockets above and close to grave soil. Previously, this process involved the tedious and expensive process of solvent extraction of soil samples. Now, a simple probe slightly thicker than a human hair can be inserted into the ground to detect decaying flesh." If you need to locate a stiff under a concrete floor, all you have to do is drill a 1/8 inch hole for probe insertion. To confirm the tester's efficacy, two NIST chemists buried dead rats and took samples over a period of 20 weeks, at which time the decomposing rodents were still detectable. No word yet as to whether they're going to look for Jimmy Hoffa. ▲

COMPUTERS AND NETWORKING

■ Origin Genesis desktop PC with optional True Fire finish.

CUSTOM GAMING COMPUTER

Late last year, some former executives of Alienware got together and formed Origin PC (**www.originpc.com**) with the aim of assembling "the best parts and technology the computer gaming industry has to offer." The result is the Genesis line of highly customized laptop, desktop, and 3D machines that you "build" from the ground up. The first job is to choose the case you want which will run you $189 to

$299, depending on your desired levels of cooling, noise, and upgradeability. Next, you pick the paint job which can push the price of the empty box up to as much as $924. From there, you choose pretty much any desired combination of processors (Intel Core i7 or i5, AMD Phenom II), memory devices, cooling techniques, and other items. After getting past the shock of the box price, things get financially less frightening. For example, if you ordered a machine with a single i7 980X 3.33 GHz processor, an ASUS Rampage III motherboard, liquid CPU cooling, 6 GB of memory, one ATI Radeon HD 5770 graphics adapter, and a pair of 1 TB drives, you'd top out at $4,215 (not including a display, keyboard, and mouse). That's a pretty significant chunk of change, but serious gaming geeks have been known to shell out much more. (For example, check out the Hardcore Reactor X at **www.hardcorecomputer.com**; it can run you close to $10,000.) ▲

DONATE YOUR SPARE CYCLES

If you're like most of us, you have a lot of computing power that sits idle most of the time. Maybe it's in the form of a PC that you've retired but haven't sold because, frankly, it isn't worth the cost of a classified ad. Or, maybe it's that 3.3 GHz multicore processor in your desktop machine that isn't exactly melting down when it fetches your daily spam. Either way, you might want to consider volunteering a bit of that processing power to a good cause. A good place to start is the University of California, Berkeley's site for volunteer and grid computing, **boinc.berkeley.edu**. In case you haven't heard of it, BOINC stands for Berkeley Open Infrastructure for Network Computing, and it's used for distributed computing in a range of diverse projects including earthquake detection, scientific and medical research, mapping of the universe, and so on. One interesting project is Einstein@Home (**einsteinathome.org**) which uses your machine to search for gravitational waves from pulsars, using data from the Laser Interferometer Gravitational-Wave Observatory (LIGO) detector. In fact, a couple months ago, volunteers from Iowa and Germany found one about 17,000 light years away. The current 500,000+ participating BOINC computers grind out almost 2,370 teraFLOPs every 24 hours, so you may as well join in the fun. All you have to do is log onto the BOINC site, choose a worthy project, and download the software. It's a rare opportunity to help create a better world while doing absolutely nothing. ▲

CIRCUITS AND DEVICES

DVD PLAYER WITH FLO TV

Most of us are eagerly awaiting the day when we can watch TV without paying exorbitant rates for cable or satellite services, and Audiovox (**www.audiovox.com**) has taken a step in that direction with its new portable DVD player with FLO TV™. The model DFL 710 offers the usual ability to play DVDs on its seven inch diagonal screen, and it includes stereo speakers. You can also get live mobile television (i.e., real-time transmission rather than downloading or buffering) via Qualcomm's FLO TV service, delivered over "America's largest dedicated mobile TV network." This brings in children's programming (Disney, Nickelodeon), live sports, news (Fox News Channel, CNN Mobile), and a range of other shows (full listing guide at **www.flotv.com**). The unit lists at $199.99 and, if you grab one quickly enough, includes three months of FLO. Thereafter, you'll need to shell out $14.99/month. ▲

■ The Audiovox DFL 710 DVD player offers live TV.

MEMORY FEATURES WATER COOLING

You can spend big bucks for a custom, overclocked computer like the ones mentioned previously, but if you have the time and inclination, it's also possible to build one yourself. In that case, the new HyperX memory modules may be of interest. Kingston Technology (**www.kingston.com**) recently introduced the HyperX H20 water-cooled DDR3 memory kits which offer frequencies up to 2,133 MHz and capacities up to 6 GB. The new line consists of three products: two 4 GB dual-channel kits

■ Kingston now offers three models of water-cooled DDR3 kits.

worldmags

(2,000 and 2,133 MHz) and a 6 GB triple-channel kit (2,000 MHz). According to Kingston's senior tech manager Mark Tekunoff, "Water cooling is desirable for its quiet operation and long-term reliability. We are bringing HyperX H2O to market as a solution for PC enthusiasts who want to build water-cooled systems." The units will run you $157, $205, or $235. ▲

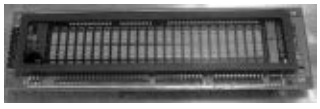
INDUSTRY AND THE PROFESSION

2.5 IMAGES COMING BY CHRISTMAS

Back in the 1960s, Ray Dolby introduced the concept of enhanced audio based on reducing noise and enhancing the desired part of the spectrum. Now Paul Darbee, CEO of DarbeeVision, Inc., has come up with a similar approach to image enhancement dubbed DARBEE Visual Presence (DVP). The process is based on experiments Darbee made using stereo videos with dual cameras. Apparently, if you defocus one image and subtract it from the remaining sharp one, you end up with a combined image that produces something that approximates a 3-D viewing experience (he refers to it as

GIZMOS THAT BRING YOUR DREAMS TO LIFE!

3 watt LED strip $10.00
Not just bright… Stupid Bright!!



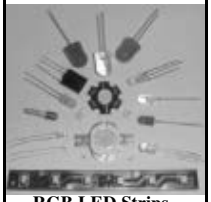
Noritake Vacuum Fluorescent Display
Regular $50 now $25.00 each
Works with all micros and arduino

Atmega 8/168/328 Development board $35

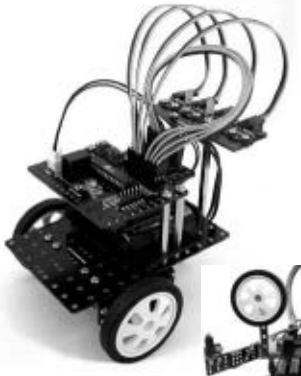
Atmel STK500 USB ISP programmer $38

Lego Flashlights
Keychain $7.95
Head Lamp $11.95
7 Inch Lamp $19.95
Dynamo Lamp $21.95
Let that light shine in the next generation!

1000's of LED's in stock
1.8mm, 3mm, 5mm, and 10mm from $.40
1/2 watt $.95
1 watt Stars $3.60
RGB $1.50
UV, IR, Surface Mount

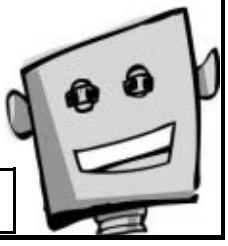


RGB LED Strips
12v LED Bars
LCD Displays
Graphic and Character



Basic robot animal kits start at $25

Robot Kits Are Here:
Completely configurable, expandable, and modifiable.
Visual programming.
Great for kids age 10 and up
With student workbook.
Starting at $95
Ask about school discounts



5797 NW Cornelius Pass Road
Hillsboro Oregon 97124

To see the best deals join our online mailing list or stop by our retail store!

www.SurplusGizmos.com
503-439-1249

Order online at:
www.melabs.com

Development Tools for PIC® MCUs
microEngineering Labs, Inc.

Phone: (719) 520-5323
Fax: (719) 520-1867
Box 60039
Colorado Springs, CO 80960

USB Programmer for PIC® MCUs
$89.95 (as shown)
RoHS Compliant
Programs PIC MCUs including low-voltage (3.3V) devices
Includes Software for Windows 98, Me, NT, XP, and Vista.

With Accessories for $119.95:
Includes Programmer, Software, USB Cable, and Programming Adapter for 8 to 40-pin DIP.

EPIC™ Parallel Port Programmer starting at $59.95

LAB-X Experimenter Boards

Pre-Assembled Board
Available for 8, 14, 18, 28, and 40-pin PIC® MCUs
2-line, 20-char LCD Module
9-pin Serial Port
Sample Programs
Full Schematic Diagram
Pricing from $79.95 to $349.95

PICPROTO™ Prototyping Boards

Double-Sided with Plate-Thru Holes
Circuitry for Power Supply and Clock
Large Prototype Area
Boards Available for Most PIC® MCUs
Documentation and Schematic
Pricing from $8.95 to $19.95

BASIC Compilers for PICmicro®

Easy-To-Use BASIC Commands
Windows 98/Me/2K/XP/Vista
PICBASIC™ Compiler $99.95
BASIC Stamp 1 Compatible
Supports most 14-bit Core PICs
Built-In Serial Comm Commands
PICBASIC PRO™ Compiler $249.95
Supports most PICmicro® MCU Families
Direct Access to Internal Registers
Supports In-Line Assembly Language
Interrupts in PICBASIC and Assembly
Built-In USB, I2C, RS-232 and More
Source Level Debugging

See our full range of products, including books, accessories, and components at:
www.melabs.com

worldmags

2.5-D). Although the result looks a bit like applying a sharpening filter in Photoshop, it goes beyond that because his images are modified so as to create highlights which your brain interprets as depth. It's a little more complicated than that, so if you want a full explanation, you'll need to visit **www.darbeevision.com**. In any event, watch for DVP to start appearing in TVs, video game consoles, digital picture frames, and so on by the end of the year. In addition, the company intends to produce "smart" HDMI cables that allow you to retrofit your current high-def TV. ▲

■ Before (left) and after modification by Darbee Vision.

ARM CONFERENCE NOVEMBER 9-11

The ARM architecture accounts for something like 90 percent of all low power embedded 32-bit RISC processors which are predominant in PDAs, cell phones, music players, and so forth. If you are involved in designing for this architecture — hardware or software — you should know about the 2010 ARM Technology Conference, scheduled for November 9-11 at the Santa Clara Convention Center. November 9 focuses on chip design issues, and November 10 and 11 are devoted to software and systems design. The conference combines classroom sessions, tutorials, panels, and product demos/exhibits. For more information, visit **www.eetimes.com/armconference**. **NV**

■ BY FRED EADY

THUMBS UP FOR THE VINCULUM-II TOOLCHAIN

The Vinculum-II embedded dual USB host controller takes all of the hard work out of interfacing and controlling USB devices. In this case the Vinculum-II goes one step further to save us both money and time. We don't have to design and fabricate a specialized printed circuit board (PCB) to put the Vinculum-II host controller to work. For about $15 plus shipping, you and I can obtain the embedded dual USB host controller IC and all of its associated resistors, capacitors, ferrite beads, and interface pins mounted on a spiffy factory-grade PCB.

A VINCULUM-II EXPANSION BOARD

Odds are that your Vinculum-II project thoughts aren't exactly the same as mine or the reader in the next town or the reader down the street. In that the FTDI folks have provided an inexpensive Vinculum-II carrier board for its 32-, 48-, and 64-pin parts, taking the time and expense to design and manufacture a generic Vinculum-II embedded dual USB host controller carrier PCB would be nothing more than reinventing the development board wheel.

I just happen to have a 64-pin variant of the Vinculum-II expansion board. The default logical and physical layout of it is drawn up in **Schematic 1**. The hardware is under the lens in **Photo 1**. The expansion board interface pins are arranged to fit perfectly into any 0.1 inch pitch perfboard. Thus, we can literally plug our expansion board into a relatively inexpensive perfboard-based embedded dual USB host controller design. No unique PCB is required.

ENOUGH OF THE VINCULUM-II HARDWARE ALREADY

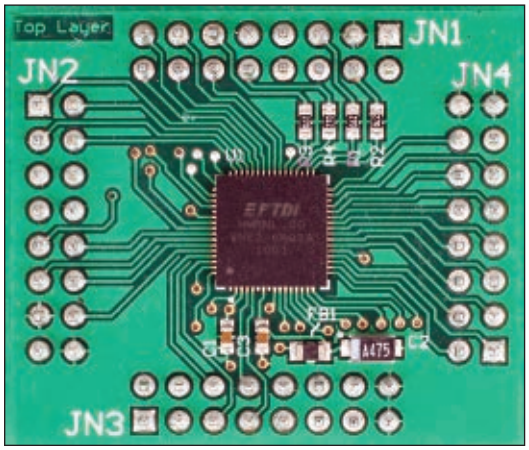
Up to this point, our Vinculum-II discussion has revolved around the hardware and the theory that stands behind it. If you've ever worked with any computing device, you know that understanding the hardware design is just as important as understanding how to talk to the intelligent hardware components via firmware. With that thought in mind, let's apply what we've learned about the Vinculum-II hardware in past Design Cycle discussions to writing some Vinculum-II firmware.

THE VINCULUM-II TOOLCHAIN

Like any other toolchain support product, the Vinculum-II toolchain begins life after a successful installation on a PC hard drive. The mainstay of our toolchain is the Vinculum-II IDE. Like most other IDEs (Integrated Development Environments), the Vinculum-II IDE allows the programmer to create projects, edit files, compile code, program devices, and debug application code.

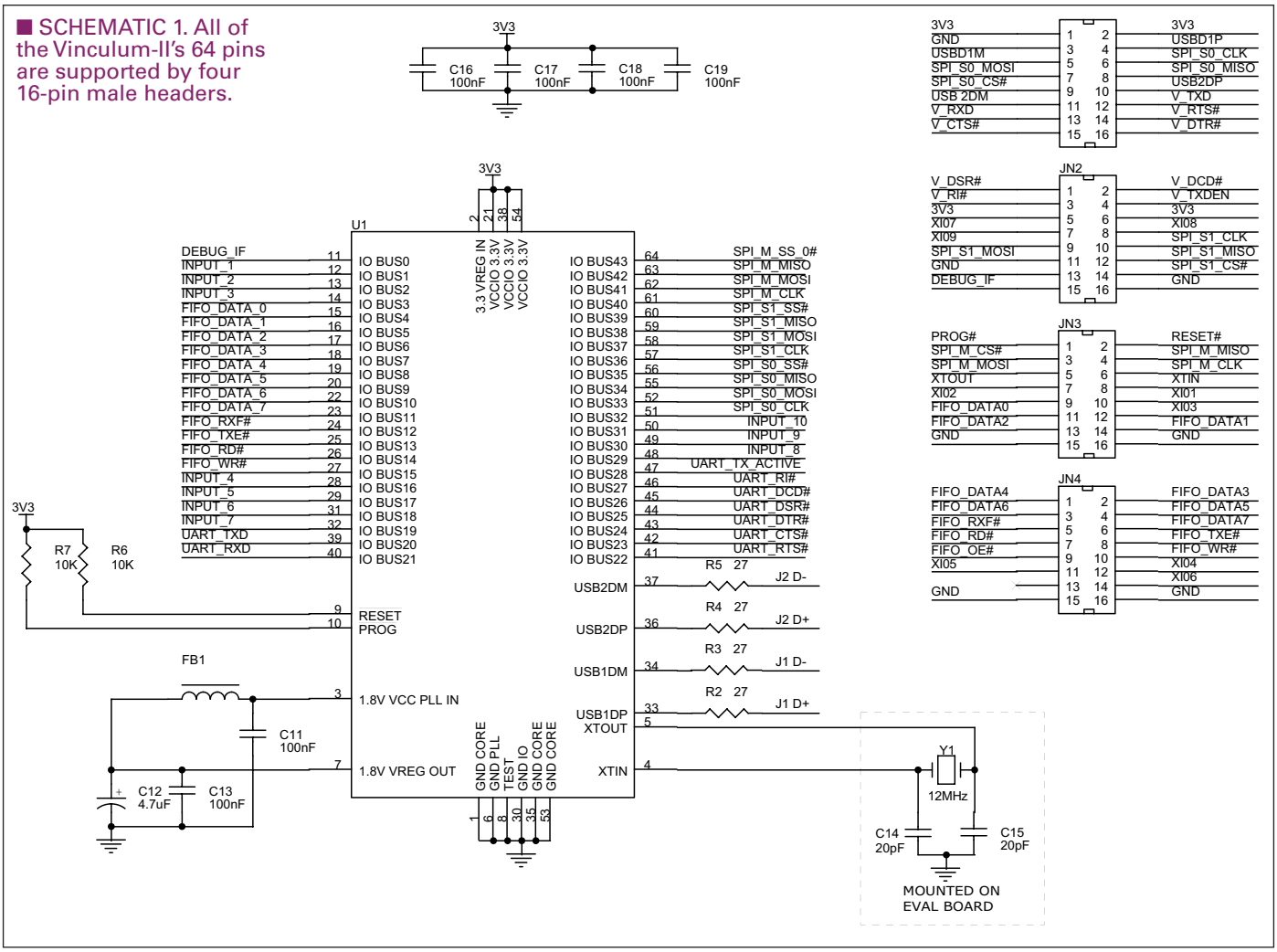
The power behind this toolchain lies in the code that forms the FTDI libraries. The FTDI libraries consist of device drivers, runtime libraries, and the VOS Kernel Services. Some of you are familiar with what device drivers and runtime libraries contribute to your code. However, you may not be privy to what the Vinculum-II VOS Kernel Services have to offer. The services provide overall application control while supplying the necessary primitives and unique data structures needed to support Vinculum-II applications.

The Vinculum-II toolchain uses an API (Application Program Interface) to access the functionality of the toolchain's device drivers and Kernal services. The API is under the control of the Device Manager which is under the command of the Kernel Services. The inclusion of a Vinculum-II API takes most of the pain out of application development



■ PHOTO 1. The Vinculum-II carrier board sits on pins that are all fitted at 0.1 inch centers which makes this carrier board perfect for integrating with standard 0.1 inch pitch perfboards.

worldmags



■ SCHEMATIC 1. All of the Vinculum-II's 64 pins are supported by four 16-pin male headers.

as the language behind the calls to the functions that reside within the device drivers is simplified. The API structure is also designed to provide the programmer easy access to the VOS Kernel Services in the same way it does for the device drivers. In most cases, the API function call names describe their actual intended function. Thus, the API calls are self commenting. I don't think I have to say a word about what these API calls do:

```
vos_dev_write
vos_dev_read
vos_delay_msecs
vos_dev_open
vos_dev_close
gpio_init
uart_init
```

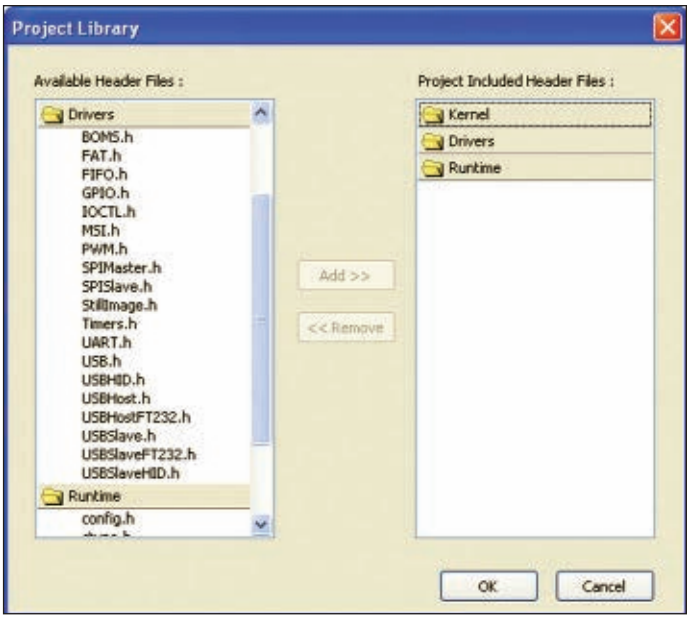
TYPICAL VINCULUM-II APPLICATION COMPONENTS

The C programming language and its syntax are used to build Vinculum-II applications. These applications usually contain a user-generated application header file, a user-generated *main* function, and user-generated threads. Application header files are not a requirement in Vinculum-II application code. However, for more complex applications, user-generated application header files are recommended as they provide a common holding point for the application's global variables and constant values.

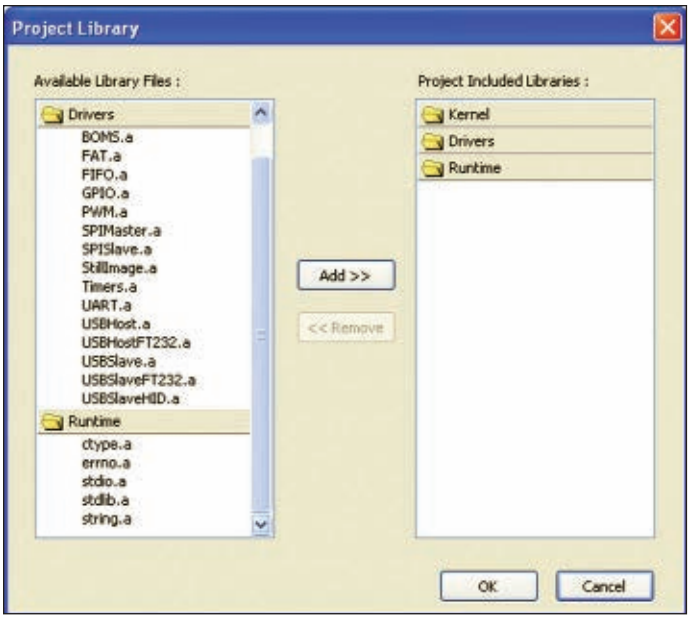
For those of you that can C already, you know that all of the initial application action is directed from the *main* function. In the case of the Vinculum-II, the threads contain the actual application functionality. The *main* function contains code that readies the system before starting the threads. The very last thing a Vinculum-II *main* function does is start the scheduler which starts the execution of the threads.

You can get a pretty good idea of the available Vinculum-II hardware resources by examining **Screenshot 1**. As you can see, the toolchain has built-in device driver header files for all of the on-chip peripheral components. There are device driver headers for file management, general-purpose I/O, SPI, timers, UARTs, and USB.

The available toolchain device driver libraries are captured in **Screenshot 2**. For the most part, there are name-associated device driver libraries that work hand in hand with the like-named device driver header files shown in **Screenshot 1**. You can add any of these header files to your project and inspect them with the IDE editor. However, you can't gain view access to the device driver



■ SCREENSHOT 1. Right-clicking on the project in the Project Manager window will reveal a drop-down menu that points you to Manage Libraries > Header Files which will get you to this point.



■ SCREENSHOT 2. You get here the same way you get to the window in Screenshot 1 by choosing Library Files instead of Header Files from the drop-down menu.

libraries via the Vinculum-II IDE editor.

Screenshot 3 reveals the VOS Kernel Services header files while **Screenshot 4** scrolls down to the Runtime headers. As you have most likely already concluded, there are library files that are associated with the header files and they can be seen in **Screenshot 5**. If a library is included in an application, its associated header file must also be included in the application.

NAVIGATING VINCULUM-II APPLICATION SOURCE CODE

VOS Kernel Services, device drivers, and API calls are major parts of the Vinculum-II application code big picture. However, we can't write an application by simply including libraries and headers into our Vinculum-II application source code. So, let's practice what we preach and begin by coding up an initial Vinculum-II application header file.

INSIDE THE APPLICATION HEADER

The application programmer guide application note states that the first thing we should include in our application header file is the size of the stack memory that the application thread will require:

```
#define SIZEOF_FIRMWARE_TASK_MEMORY 0x1000
```

The stack memory size of 0x1000 is overkill by design. After successfully compiling, loading, and running the application we're about to discuss and build with the 0x1000 value, I was able to get the application to run reliably with a stack size of only 0x0400. Thus, the stack memory size value is dependent on the complexity of your application.

The next recommended programming action is to fix a number of devices that the application will use:

```
#define NUMBER_OF_DEVICES    5
#define VOS_DEV_USB_HOST1    0
#define VOS_DEV_BOMS         1
#define VOS_DEV_USB_HOST2    2
#define VOS_DEV_UART         3
#define VOS_DEV_GPIO         4
```

Each device we list must have a unique device identifier which is used later by the Device Manager. If you're having problems with the pair of HOST definitions, recall that the Vinculum-II IC has a pair of USB portals. These portals can be identified in **Schematic 1** as USB1 and USB2. The last entry in our application header file is a forward declaration to the user-generated application thread which is really no more than a C function. A forward declaration is coded for each thread in the application. In our case, we only have one:

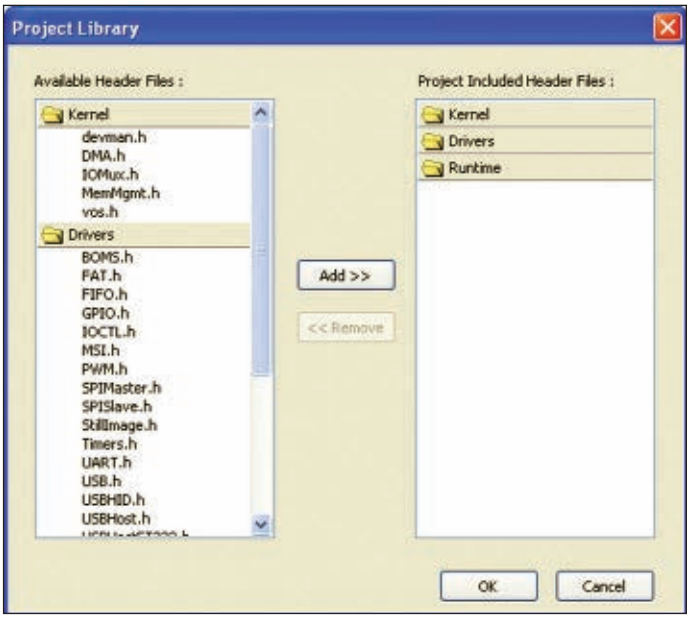
```
void application_thread(void);
```

With the application-specific header file code completed, we can move on into the application's source code file which we will call *DesignCycle-App.c*. Naturally, the application header file we just coded is named *DesignCycle-App.h*. Our first order of business is standard C fare. We will offer sacrifices unto the *#include* gods:

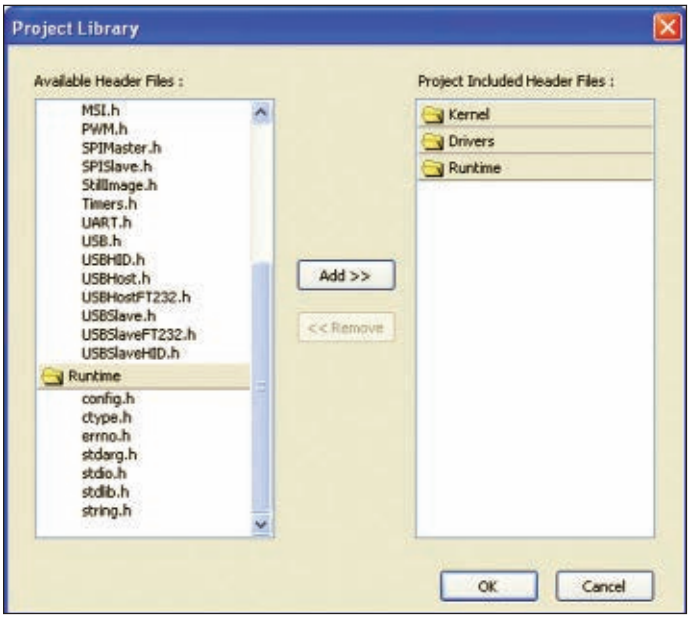
```
#include "vos.h"
#include "devman.h"
#include "DMA.h"
#include "IOMUX.h"
```

You will recognize the aforecoded *#include* statement arguments as members of the resident Kernel header files (see **Screenshot 3**). The VOS Kernel Services header file *vos.h* is a must-have as it supports the Kernel library which is the core power behind our application. Our application *#include* list acts as a guide for the device driver files we

worldofpcbs



■ SCREENSHOT 3. Focus on the Kernel header files in this shot. The purpose is to give you an idea of the division of the VOS Kernel Services and the device drivers which are under the control of the Device Manager.



■ SCREENSHOT 4. The Runtime header files are the same header files you'll find in most any C program development environment.

need to include in the application source code space:

```
#include "USBHost.h"
#include "USB.h"
#include "MSI.h"
#include "BOMS.h"
#include "UART.h"
#include "FAT.h"
#include "GPIO.h"
#include "string.h"
#include "DesignCycle-App.h"
```

The *MSI.h* and *FAT.h* files support the *BOMS.h* functionality. MSI is short for Mass Storage Interface while FAT is the good old Microsoft acronym that translates to File Allocation Table. BOMS — Bulk Only Mass Storage — is a USB class that describes a device for communicating with mass storage devices. In the Vinculum-II world, a mass storage device doesn't physically get any bigger than a thumb drive.

Once a device is opened, Device Manager returns a unique handle for that device. The returned handle is of the type VOS_HANDLE. Here's how VOS_HANDLE is declared within the Kernel's *devman.h* file:

```
#define VOS_HANDLE       uint16

VOS_HANDLE vos_dev_open(uint8 dev_num);
```

VOS_HANDLE is simply an unsigned 16-bit integer that holds the unique handle value returned by the *vos_dev_open* function which is one of the Device Manager functions. According to the device list coded in the application header, we need to reserve five unique VOS_HANDLE slots:

```
VOS_HANDLE    hUsb1, hUsb2, hUart, hBoms, hGpio;
```

The FAT driver needs to have a context declared to allow it to communicate with the file system on our mass storage device. Here's how the FTDI folks do it:

```
fat_context      fatContext;
```

And, that's how we will do it. The *fat_context* code that follows is found within the *FAT.h* header file. The FAT driver is layered on top of the BOMS driver. Thus, the context declaration is a bit different from context code you'll be exposed to as we continue:

```
// context pointer for instance of FAT
// file system
typedef void *fat_context;

fat_context *fat_open(VOS_HANDLE hBoms, unsigned
char partition, unsigned char *status);
```

According to the *fat_context* source code we can gather from the various Vinculum-II toolchain header files, it seems that the FAT context data is derived from the invocation of the *fat_open* function. The *fat_close* function adds fuel to that fire as the returned *fat_context* information is also used by the *fat_close* function:

```
void fat_close(fat_context *fat_ctx);
```

The *vos_create_thread* function returns a pointer to the newly registered thread. So, we'll need to accommodate that action by declaring a pointer to our user-generated application thread which we will name *application_thread*:

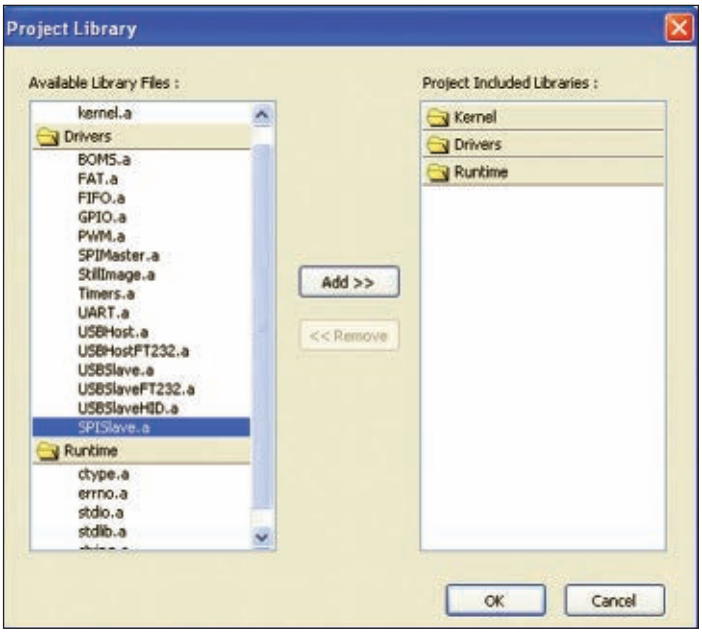
```
vos_tcb_t        *tcbApplication_thread;
```

Although the thread is not created until the last moment in the *main* function, I'll provide an advanced look at the thread creation function call so you can relate the **tcbApplication_thread* pointer we declared to the thread creation process:

```
tcbApplication_thread = vos_create_thread
(29, SIZEOF_FIRMWARE_TASK_MEMORY,
application_thread, 0);
```

The first parameter in the *vos_create_thread* function

worldmags



■ SCREENSHOT 5. Every header file that is directly associated with a like-named library file contains definitions and data structures to support the functions contained within the library file it supports.

(29) is the thread priority. A priority of 31 is highest with a priority of one being lowest. Recall that we declared the SIZEOF_FIRMWARE_TASK_MEMORY value in our application header file. We also coded a forward reference to the user-generated application thread (*application_thread*) we are creating. The thread *application_thread* is not coded to accept any arguments which explains the zero at the end of the thread creation function.

INSIDE THE MAIN FUNCTION

It's time to put all of those plans we made in the application header file into action. Let's begin by declaring a context for the USB host. Before we actually code the host context declarations, let's examine the code structure behind a USB host context. All we have to do is pull up the USBHost.h file in the Vinculum-II IDE:

```
// Context for USB Host
typedef struct _usbhost_context_t {
        // number of interfaces both USB
        // hosts combined
        unsigned char if_count;
        // number of endpoints (excluding control
        // endpoints) expected
        unsigned char ep_count;
        // number of concurrent transaction
        // expected
        unsigned char xfer_count;
        // number of concurrent isochronous
        // transactions expected
        unsigned char iso_xfer_count;
} usbhost_context_t;
```

If you're an avid Design Cycle reader, this USB host code is no stranger to you. The *typedef struct* source code snippet defines a structure of type *usbhost_context_t*. We can create instances of the structure type *usbhost_context_t* like this:

```
void main(void)
{
     // USB Host configuration context
     usbhost_context_t usb_ctx;
```

The newly declared USB host configuration context is called *usb_ctx*. The UART context can be declared in an identical manner. Here's what the UART context structure looks like from inside of the *UART.h* file:

```
// Context for UART
typedef struct _uart_context_t {
     unsigned char buffer_size;
} uart_context_t;
```

An instance of the *uart_context_t* structure called *uart_ctx* is created within the confines of the *main* function in this manner:

```
// UART configuration context
uart_context_t uart_ctx;
```

I think you can get there from here as far as the GPIO context is concerned. Here's the GPIO seed context structure definition:

```
// GPIO context
typedef struct _gpio_context_t {
     unsigned char port_identifier;
} gpio_context_t;
```

And ... here's the code that declares an instance of the *gpio_context_t* structure type:

```
// GPIO configuration context
gpio_context_t gpioCtx;
```

The common element of all of the context structures is configuration data that pertains to the device the structure supports. Each device context is used to set the configuration of a device before it is opened.

Now that we've voiced our intentions to support, configure, and possibly run five devices, let's prime the pump by initializing the VOS Kernel Services. For clarity, I've taken the liberty to pull the *vos_init* parameter values from their respective header files and place them with the *vos_init* function call:

```
#define NUMBER_OF_DEVICES    5

//*********************************************
// VOS INITIALIZATION
//*********************************************

// VOS initialization and start-up
void vos_init(uint8 quantum, uint16 tick_cnt,
uint8 num_devices);
void vos_start_scheduler(void);

// default interval for timer interrupts
#define VOS_TICK_INTERVAL     1

// default time-slice quantum for tasks in
RUNNING state
#define VOS_QUANTUM           50

vos_init(10, VOS_TICK_INTERVAL,
NUMBER_OF_DEVICES);
```

The VOS_QUANTUM default value has been

worldofpcbs

overridden with a value of 10 decimal. The VOS_xxxx definitions were gleaned from the *vos.h* header file. We entered the NUMBER_OF_DEVICES value when we coded the application header file *DesignCycle-App.h*.

This would be a good time to set up the Vinculum-II clocking. As you might imagine, the toolchain API has a call for that:

```
vos_set_clock_frequency(VOS_48MHZ_CLOCK_
FREQUENCY);
```

As you can see in **Schematic 1**, the 48 MHz clock is derived from a 12 MHz crystal and the Vinculum-II's 4x PLL. In a previous Design Cycle Vinculum-II discussion, we took a detailed look at a helper program called the VNC2 IOMux Config utility which is part of the Vinculum-II toolchain package. If you had the opportunity to join in on that conversation, you'll recall that the VNC2 IOMux Config utility is used to configure the GPIO subsystem. An added feature of this utility is that it writes the GPIO configuration code for us. Well, we need some UART I/O code for sure and if we want to add some status LEDs to the mix, we should go ahead and lay out some output I/O pins to support them, as well. Here's what the VNC2 IOMux Config utility produced according to my idea of where things should go:

```
// GPIO port A bit 1 to pin 12 - LED3 ON
// EVAL BOARD
vos_iomux_define_output(12,IOMUX_OUT_GPIO_PORT_
A_1);
// GPIO port A bit 2 to pin 13 - LED4 ON
// EVAL BOARD
vos_iomux_define_output(13,IOMUX_OUT_GPIO_PORT_
A_2);
// GPIO port A bit 5 to pin 29 - LED5 ON EVAL
// BOARD
vos_iomux_define_output(29,IOMUX_OUT_GPIO_PORT_
A_5);
// GPIO port A bit 6 to pin 31 - LED6 ON EVAL
// BOARD
vos_iomux_define_output(31,IOMUX_OUT_GPIO_PORT_
A_6);
// UART to V2EVAL board pins
vos_iomux_define_output(39,IOMUX_OUT_UART_TXD);
 //UART Tx
vos_iomux_define_input(40,IOMUX_IN_UART_RXD);
 //UART Rx
vos_iomux_define_output(41,IOMUX_OUT_UART_RTS_N);
 //UART RTS#
vos_iomux_define_input(42,IOMUX_IN_UART_CTS_N);
 //UART CTS#
```

I'm sure you're wondering why I chose these particular I/O pins and LED identifiers. The cat is let out of the bag in the comments area of the code generated by the VNC2 IOMux Config utility. I guess it would be a good idea to add some LED definitions to our application header code:

```
#define LED0    0x02   //LED3 ON EVAL BOARD
#define LED1    0x04   //LED4 ON EVAL BOARD
#define LED2    0x20   //LED5 ON EVAL BOARD
#define LED3    0x40   //LED6 ON EVAL BOARD
```

It is impossible for us to produce a single PCB and purchase all of the necessary electronic components to

WIRELESS MADE SIMPLE®
QUICKLY AND COST-EFFECTIVELY MAKE YOUR PRODUCT WIRELESS
RF MODULES
LOW COST • LONG RANGE
TRANSMITTERS • RECEIVERS
•TRANSCEIVERS
INTEGRATED CIRCUITS
ENCODERS • DECODERS • TRANSCODERS
COMMAND • CONTROL
•KEYLESS ENTRY
OEM PRODUCTS
EASILY CUSTOMIZED
ANTENNAS
FROM WHIPS TO CHIPS
Featured Products:
LT Series Transceiver
•315 MHz
•418 MHz
•433 MHz
• Data Rates to 10K bps
• Direct Serial Interface
• Low Power Consumption
• Low Cost
• Long Range
• Ultra Compact
Ideal for the bi-directional wireless transfer of serial data, control, or command information in the favorable 260-470MHz band. Its advanced synthesized architecture delivers outstanding stability and frequency accuracy.
Function Relay Module
Provides four latched or momentary relay outputs that are capable of switching external AC or DC powered loads of up to 5 amps.
•315 MHz
•418 MHz
•433 MHz
• Dual Power Inputs
• 256 Unique Addresses
• Wide Operational Voltage
• FCC and Industry Canada Pre-Tested
LINX TECHNOLOGIES
800-736-6677
159 Ort Lane
Merlin, OR 97532
linxtechnologies.com

worldmags

Vinculum VNC2

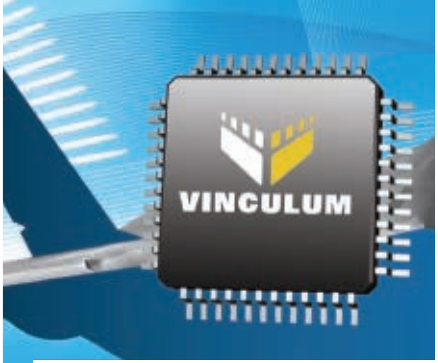
SPEED.
FLEXIBILITY.
PERFORMANCE.

A programmable system-on-chip
USB 2.0 Host / Slave controller

- Dual channel USB 2.0 interface, handles all USB host and data transfer functions in single IC.
- On-chip 16-bit Harvard architecture MCU core with 256 Kbyte Flash and 16kbyte RAM.
- External UART, FIFO, SPI Slave, SPI Master, GPIO and PWM interfaces.
- Vinculum-II software development tools available for user application development.
- Multiple package size options including VNC1L backwards compatible package option.
- Targeted for range of USB applications, from portable media devices and cell phones to industrial and automotive applications.

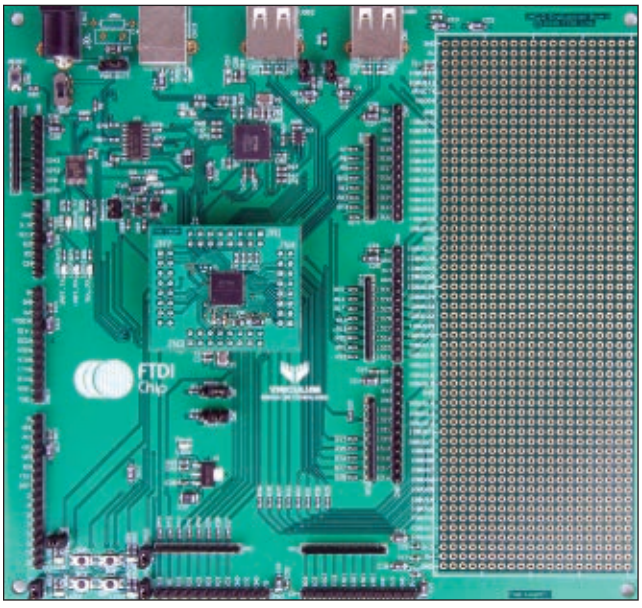
Vinculum-II evaluation modules

- V2DIP1/2 - Miniature VNC2 Development Module with Single or Dual USB Connectors
- V2-EVAL - Complete Evaluation & Development Kit for VNC2
- VNC2 Debug Module



USB MADE EASY
www.ftdichip.com

FTDI Chip



■ PHOTO 2. Every pin of the Vinculum-II Expansion Board is terminated at a header point or the prototype area. All of this USB horsepower is yours for about 100 bucks!

support the Vinculum-II Expansion Board for less than $80. So, rather than invent the development board wheel yet again, it is more cost-effective to simply purchase the factory-produced FTDI Vinculum-II Evaluation Board for $79. **Photo 2** shows our board mounted on a Vinculum-II Evaluation Board.

Thus far, all of our Vinculum-II application code has targeted the Evaluation Board peripheral layout. With that, let's move on and continue bringing the Board peripherals to life. The USBHost context contains a field that holds the maximum number of interfaces to support (*if_count*). We'll use the *if_count* value and the rest of the USBHost context to initialize the USBHost interfaces:

```
usb_ctx.if_count = 2;
usbhost_init(VOS_DEV_USB_HOST1, VOS_DEV_
USB_HOST2, &usb_ctx);
```

The following sequence of events wraps up the actions initiated inside of the *main* function. The BOMS service is initialized, a UART buffer is allocated, the UART is initialized, the GPIO port is initialized, the application thread is created, and the scheduler is started which allows the thread code to be executed:

```
boms_init(VOS_DEV_BOMS);

uart_ctx.buffer_size = VOS_BUFFER_SIZE_128_BYTES;
uart_init(VOS_DEV_UART, &uart_ctx);
gpioCtx.port_identifier = GPIO_PORT_A;
gpio_init(VOS_DEV_GPIO,&gpioCtx);

tcbApplication_thread = vos_create_thread(29,
SIZEOF_FIRMWARE_TASK_MEMORY,          application_thread, 0);

vos_start_scheduler();

main_loop:
goto main_loop;
}
```

The scheduler will continually queue the application thread for execution as the main function turns in a very tight loop.

THINGS ASSOCIATED WITH APPLICATION THREADS

Lots of device open and iocb (I/O Control Block) operations are the norm for an application thread. For instance, here is the sequence of events to set up the GPIO to support the LEDs:

```
// THIS CODE FOUND IN GPIO.h
#define VOS_IOCTL_GPIO_SET_MASK  1
   // set pins to either input (0) or output (1)

// GPIO control block for use with GPIO IOCTL
```


worldmags

```
// function
typedef struct _gpio_ioctl_cb_t {
     unsigned char ioctl_code;
     unsigned char value;
} gpio_ioctl_cb_t;

     //THIS CODE FOUND IN application_thread
     unsigned char leds;
     gpio_ioctl_cb_t gpio_iocb;
     hGpio = vos_dev_open(VOS_DEV_GPIO);
     gpio_iocb.ioctl_code = VOS_IOCTL_GPIO_
     SET_MASK;
     gpio_iocb.value = 0xff;
          // set all as output
     vos_dev_ioctl(hGpio, &gpio_iocb);
```

I pulled the SET_MASK and GPIO control block definitions from the GPIO.h file and added them in the code mix for clarity. Once the *gpio_iocb* control block is instantiated, it gets used as a tool to set the I/O data direction. Just before the I/O pin data direction is set, the GPIO device is opened and its handle is returned to *hGpio*. The Vinculum-II Evaluation Board LEDs are all connected as common anode with respect to the power source. Thus, the Vinculum-II drives the cathodes from its I/O pins. These lines of code will illuminate all of the LEDs we've defined:

```
leds = 0;
vos_dev_write(hGpio,&leds,1,NULL);
```

These lines of code will extinguish all of the defined LEDs:

```
leds = 0xFF;
vos_dev_write(hGpio,&leds,1,NULL);
```

Want to turn off LED0 only? Here's how we do it:

```
leds = LED0;
vos_dev_write(hGpio,&leds,1,NULL);
```

The *vos_dev_write* API call uses the GPIO handle (hGpio) to identify the GPIO pins that are driving the LEDs and writes the value of *leds* to the I/O port. Pretty clever, huh?

Fred Eady can be reached at fred@edtp.com.

SOURCES

FTDI
Vinculum-II Embedded Dual USB Host Controller
Vinculum-II IDE
Vinculum-II Toolchain
www.ftdichip.com

THUMB DRIVE HEAVEN

Communicating with a thumb drive is just as easy as controlling the bank of LEDs. The FTDI Vinculum-II site contains detailed code examples of how to put a thumb drive online, as well as the Vinculum-II toolchain and a treasure trove of application notes. You've all been asking me to deliver low cost projects with a punch. I think I have succeeded. For about 100 bucks you can go to thumb drive heaven and add the Vinculum-II embedded dual USB host controller to your Design Cycle. **NV**

Over 1.7 Million Parts Online

IMagine the Possibilities.

Scan with your smart phone or mobile device's camera/ QR-Code reader.
mouser.com

WARNING: Designing with Hot, New Products May Cause A Time-to-Market Advantage.

MOUSER ELECTRONICS
a tti company

mouser.com (800) 346-6873

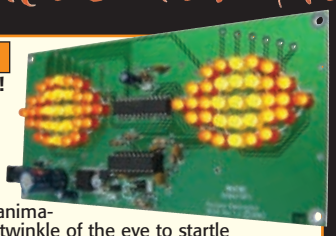
Mouser and Mouser Electronics are registered trademarks of Mouser Electronics, Inc. Other products, logos, and company names mentioned herein, may be trademarks of their respective owners.

worldmags

RAMSEY

Spooktacular Halloween Treats!

Blinky-Eyes Animated Display



- ✔ **Animated display of 66 super bright LED's!**
- ✔ **Microcontroller controlled!**
- ✔ **Changes brightness automatically!**
- ✔ **Animated with constant motion!**

The ultimate animated LED kit that will dazzle you and delight your friends! Uses a microcontroller to randomly select from many different animations such as a long pause before a wink, or a twinkle of the eye to startle passers-by!

Four modes to satisfy any enthusiast's desires: 1. Off for long random periods, then blinks or winks. Designed to scare! 2. On for long periods before performing an animation, perfect for costumes and displays! 3. Animates all the time for constant motion, perfect display attention-getter. 4. Random fire! When placed in a pumpkin will light it up like you wouldn't believe! As if this weren't enough, the BE66 can also control a small hobby motor to shake bushes at random intervals or signal an external player to make a scary sound! Also has a CDS cell to sense light. In one mode, the display will dim as it gets dark for battery operation, and in the other it will turn off when it's too bright, so it plays only in the dark!

BE66 **Blinky-Eyes Animated Display Kit** **$59.95**

Laser Light Show

- ✔ **Audio input modulates pattern!**
- ✔ **Adjustable pattern & size!**
- ✔ **Projects neat motorized patterns!**
- ✔ **Uses safe plastic mirrors!**

You've probably seen a laser show at concerts or on TV. They're pretty impressive to say the least! Knowing that you can't afford a professional laser display we challenged our engineers to design one that's neat and easy to build, yet inexpensive.

Well, the result is the LLS1 Laser Light Show! This thing is sweet and perfect for your haunted house or halloween parties! It utilizes two small motors and a small standard laser pointer as the basics. Then, we gave it variable pattern and speed controls to customize the pattern!

Not enough, you say? How about a line level audio input to modulate the pattern with your CD's, music, or spooky sound effects? You bet! Everything is included, even the small laser pointer. Runs on 6-12 VDC or our standard AC adapter.

LLS1 **Laser Light Show Kit** **$49.95**
AC125 **110VAC Power Supply** **$9.95**

Tri-Field Meter & *"Ghost Detector"*

- ✔ **See electric, magnetic, and RF fields!**
- ✔ **Watch the magnetic fields of the earth!**
- ✔ **Sense different magnetic poles**
- ✔ **Detect RF transmitter fields**
- ✔ **Graphical LED display allows you to "see"the invisible fields**
- ✔ **Great learning tool for EMF, RF, and manetic field theory.**

Call it a Tri-Field Meter, an Electrical, Magnetic, and RF Detector, a Ghost Detector, or a Tricorder, but walking around with this on Halloween will seriously impress even the most doubting!

The TFM3C has three separate field sensors that are user selectable to provide a really cool readout on two highly graphical LED bargraphs! Utilizing the latest technology, including Hall Effect sensors, you can walk around your house and actually "SEE" these fields around you! You will be amazed at what you see. How sensitive is it? Well, you can see the magnetic field of the earth... THAT'S sensitive!

The technical applications are endless. Use it to detect radiation from monitors and TV's, electrical discharges from appliances, RF emissions from unknown or hidden transmitters and RF sources, and a whole lot more! If you're wondering whether your wireless project or even your cell phone is working, you can easily check for RF! A 3-position switch in the center allows you to select electric, magnetic, or RF fields. A front panel "zero adjust" allows you to set the sensors and displays to a known clean "starting point."

If the TFM3C looks familiar, it's probably because you saw it in use on the CBS show Ghost Whisperer! It was used throughout one episode (#78, 02-27-2009) to detect the presence of ghosts!

The concept is simple, it is believed (by the believers!) that ghosts give off an electric field that can be detected with the appropriate equipment. In the electric mode, the TFM3C's displays will wander away from zero even though there isn't a clear reason for it (not scientifically explainable, aka paranormal!). This would mean something has begun to give off an electric field. What it was in the Ghost Whisperer was a friendly ghost. What it will be in your house... who knows! Makes a GREAT leaning project besides! Requires 4 AA batteries.

TFM3C **Tri-Field Meter Kit With Case** **$74.95**

Automatic Animated Ghost

- ✔ **Automatically greets your visitors!**
- ✔ **Responds to sudden noises!**
- ✔ **Built-in microphone!**
- ✔ **Adjustable sensitivity**

Who says ghosts are make believe? Once your friends come upon this one they'll think differently! The unique circuit board design includes two ominous blinking eyes that change with various conditions, including sudden changes in ambient noise. A highly sensitive built-in microphone picks up anything from noises to talking and makes the ghost dance with its built-in motor, make eerie sounds with the built-in speaker, and randomly blink. A white cloth and a hanger are included as shown to make it look like the real thing. Runs on 2 AAA batteries (Not included).

MK166 **Automatic Animated Ghost Kit** **$21.95**

Halloween Pumpkin

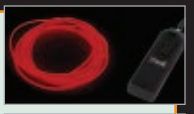
- ✔ **25 bright LED's!**
- ✔ **Random flash simulates flickering candle!**
- ✔ **Super bright LED illuminates entire pumpkin!**
- ✔ **Simple & safe 9V battery operation**

The perfect "starter" kit with a terrific Halloween theme! You won't be scraping the seeds and guts out of this pumpkin! Six transistor circuit provides a neat random flash pattern that looks just like a flickering candle. Then a super bright LED illuminates the entire pumpkin with a spooky glow!

The pumpkin face is the actual PC board, and assembly is easy through-hole soldering of all components and LED's. Your pumpkin is powered by a standard 9V battery (not included) which snaps to the back of the pumpkin. An on/off switch is also included. Create a new kind of pumpkin this year, and learn about LED's and electronics at the same time!

MK145 **Electronic Halloween Pumpkin Kit** **$10.95**

Portable EL Electroluminescence



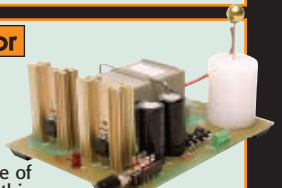
- ✔ **3.3 feet long!**
- ✔ **Low power consumption!**
- ✔ **Highly visible brilliant colors**

Electroluminescence illuminated flexible wire sets can be used for a lot of things but when they're battery powered they're perfect for Halloween and Trick or Treat safety concerns! Each thin, flexible EL wire set is 3.3' long and runs on two standard AA batteries (not included). Current consumption is less than 100mA for long life.

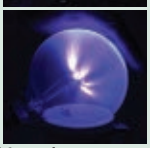
Mode settings include steady glow and slow or fast flash! Make it part of a brilliantly lit, custom designed costume or simply add it for illuminated safety while Trick or Treating in the dark.

NWRR15 **EL Illumination Wire Set, Red** **$19.95**
NWRG15 **EL Illumination Wire Set, Green** **$19.95**
NWRB15 **EL Illumination Wire Set, Blue** **$19.95**

Spark Generating HV Plasma Generator



- ✔ **Generate 2" sparks to a handheld screwdriver!**
- ✔ **Light fluorescent tubes without wires!**
- ✔ **Build your own Plasma Balls!**
- ✔ **Generates up to 25kV @ 20 kHz from a solid state circuit!**



This popular kit was conceived by one of our engineers who likes to play with things that can generate large, loud sparks, and other frightening devices! And at Halloween there's no better effect than high voltage sparks flying through the air! The PG13 Plasma Generator creates a very impressive 25,000 volts at 20 kHz, to provide a stunning display of high voltage! It will draw a cool looking 2" spark to hand held screwdriver, or light fluorescent tubes without any connection!

It produces stunning lighting displays, drawing big sparks, to perform lots of high voltage experiments. In the picture, we took a regular clear "Decora" style light bulb and connected it to the PG13 - WOW! A storm of sparks, light tracers and plasma filled the bulb. Holding your hand on the bulb doesn't hurt a bit and you can control the discharge! It can also be used for powering other experiments; let your imagination be your guide! Operates on 16VAC/VDC for maximum output. Can also be run from 5-16VAC/VDC to reduce the output voltage.

PG13 **Plasma Generator Kit** **$64.95**
PS21 **110VAC Input, 16VAC Output, Power Supply** **$19.95**

worldmags

OBDII *CarChip Pro*

The incredible OBDII plug-in monitor that has everyone talking! Once plugged into your vehicle it monitors up to 300 hours of trip data, from speed, braking, acceleration, RPM and a whole lot more. Reads and resets your check engine light, and more!

8226 CarChip Pro OBDII Monitor $99.95

Practice Guitar Amp & DI

Practice your guitar without driving your family or neighbors nuts! Works with any electric, acoustic-electric, or bass guitar. Plug your MP3 player into the aux input and practice to your favorite music! Drives standard headphones and also works as a great DI!

PGA1 Personal Practice Guitar Amp Kit $64.95

Passive Aircraft Monitor

The hit of the decade! Our patented receiver hears the entire aircraft band without any tuning! Passive design has no LO, therefore can be used on board aircraft! Perfect for airshows, hears the active traffic as it happens! Available kit or factory assembled.

ABM1 Passive Aircraft Rcvr Kit $89.95

LED Blinky

Our #1 Mini-Kit for over 35 years! Alternately flashes two jumbo red LED's. Great for signs, name badges, model railroading, and more. Used throughout the world as the first learning kit for students young and old! Great solder practice kit. Runs on 3-15 VDC.

BL1 LED Blinky Kit $7.95

LED SMT Blinky

The subminiature high-tech version of the BL1! Utilizes all SMT components to drive a pair of high intensity SMT LED's. Runs on 2 button cells (included). Great attention grabber for signs, hats, or other applications.

BL2 LED SMT Blinky Kit $17.95

Electronic Siren

Exactly duplicates the upward and downward wail of a police siren. Switch closure produces upward wail, releasing it makes it return downward. Produces a loud 5W output, and will drive any speaker! Horn speakers sound the best! Runs on 6-12VDC.

SM3 Electronic Siren Kit $7.95

Universal Timer

Build a time delay, keep something on for a preset time, provide clock pulses or provide an audio tone, all using the versatile 555 timer chip! Comes with circuit theory and a lots of application ideas and schematics to help you learn the 555 timer. 5-15VDC.

UT5 Universal Timer Kit $9.95

Voice Activated Switch

Voice activated (VOX) provides a switched output when it hears a sound. Great for a hands free PTT switch or to turn on a recorder or light! Directly switches relays or low voltage loads up to 100mA. Runs on 6-12 VDC.

VS1 Voice Switch Kit $9.95

Tone Encoder/Decoder

Encodes OR decodes any tone 40 Hz to 5KHz! Add a small cap and it will go as low as 10 Hz! Tunable with a precision 20 turn pot. Great for sub-audible "CTS" tone squelch encoders or decoders. Drives any low voltage load up to 100mA. Runs on 5-12 VDC.

TD1 Encoder/Decoder Kit $9.95

20 Watt Mini Audio Amp

Delivers a super clean 20W output from one SMT package! Ultra efficient class D design produces no heat. PCB can be snapped into a small circle for special applications. Runs on 18VDC for rated output, or down to 10VDC for reduced output.

UAM2 20W Subminiature Amp Kit $34.95

Touch Switch

Touch on, touch off, or momentary touch hold, it's your choice with this little kit! Uses CMOS technology. Actually includes TWO totally separate touch circuits on the board! Drives any low voltage load up to 100mA. Runs on 6-12 VDC.

TS1 Touch Switch Kit $9.95

Walking Electronic Bug

Built around a pair of subminiature cell phone motors, this bug wanders around looking for things to bump into! Sensors below his LED eyes sense proximity and make him turn away! Steer him with flashlights too! Runs on two "N" batteries.

WEB1 Walking Bug Kit $29.95

Mad Blaster Warble Alarm

If you need to simply get attention, the "Mad Blaster" is the answer, producing a LOUD ear shattering raucous racket! Super for car and home alarms as well. Drives any speaker. Runs on 9-12VDC.

MB1 Mad Blaster Warble Alarm Kit $9.95

Xenon Tube Strobe Light

Create amazing effects with an authentic Xenon tube strobe light! Creates a super bright white FLASH with a variable speed of 2 to 20 flashes second. Just connect 110VAC and and you have a complete strobe light!

K5300 Xenon Tube Strobe Light Kit $19.95

Stereo Ear Super Amplifier

Ultra high gain amp boosts audio 50 times and it does it in stereo with its dual directional stereo microphones! Just plug in your standard earphone or headset and point towards the source. Incredible gain and perfect stereo separation!

MK136 Stereo Ear Amp Kit $9.95

Water Sensor Alarm

This little $8 kit can really "bail you out"! Simply mount the alarm where you want to detect water level problems (sump pump!). When the water touches the contacts the alarm goes off! Sensor can even be remotely located. Runs on a standard 9V battery.

MK108 Water Sensor Alarm Kit $6.95

Air Blasting Ion Generator

Generates negative ions along with a hefty blast of fresh air, all without any noise! The steady state DC voltage generates 7.5kV DC negative at 400uA, and that's LOTS of ions! Includes 7 wind tubes for max air! Runs on 12-15VDC.

IG7 Ion Generator Kit $64.95

Retro Nixie Tube Clock

Genuine Nixie tubes popular in the 50's brought back in one of the neatest digital clocks around today! Hand made teak maple base, 12/24 hour format, soft fade-out, auto-dim, and a crystal time base at 20ppm! Tube kits also available.

IN14TM HV Plasma Generator Kit $329.95

SMT Soldering Lab

Learn all about SMT parts and how to solder them! Surface mount parts are tiny and require a special skill to solder. This lab and course covers it all, and you end up with a great "Decision Maker" kit when done!

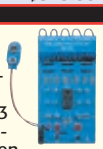
SM200K SMT Soldering Lab Kit $22.95

Soldering Lab

The perfect beginner's project specifically designed to teach you the fundamentals of soldering and PC boards. You will not only learn soldering, but hoto troubleshoot soldering problems and how to fix them! Final project runs on 9V battery.

SP1A Soldering Lab Kit $9.95

Soldering Parts Lab



Not only will you learn soldering and desoldering but you will also learn component identification and color codes! Includes 163 solder points, and you will learn how to solder all types of components. The comprehensive manual even includes exams!

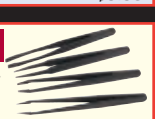
SP3B Soldering Parts Lab Kit $9.95

IC AM/FM Radio Lab

Learn all about AM/FM radio theory, IC theory, and end up with a high quality radio! Extensive step-by-step instructions guide you through theory, parts descriptions, and the hows and whys of IC design. Runs on a standard 9V battery.

AMFM108K AM/FM IC Radio Lab Kit $34.95

Non-Conductive Tweezers



We've discussed tweezers and magnifiers while working with SMT components. Working with highly sensitive components, ESD safe tweezers can be a life saver! This set of 4 non-conductive tweezers are perfect for any static sensitive devices, and are priced right!

VTTWSET2 Non-Conductive Tweezers Set $3.95

3-In-1 Multifunction Lab

The handiest item for your bench! Includes a RoHS compliant temp controlled soldering station, digital multimeter, and a regulated lab power supply! All in one small unit for your bench! It can't be beat!

LAB1U 3-In1 Multifunction Solder Lab $129.95

Get The Catalog!

Get the latest 2010 Ramsey Hobby Catalog! ***96 value packed pages*** of the neatest goodies around with lots of new stuff! Order yours today on line or give us a call... Or download the PDF at www.ramseykits.com/catalog!

ramsey amazon.com ebay

www.ramseykits.com

800-446-2295

ramsey
www.ramseykits.com

590 Fishers Station Drive
Victor, NY 14564
(800) 446-2295
(585) 924-4560

Where Electronics Is Always Fun!

✦ ***Build It!*** ✦ ***Achieve It!***
✦ ***Learn It!*** ✦ ***Enjoy It!***

Prices, availability, and specifications are subject to change. Not responsible for typos, stupids, printer's bleed, or really weird Halloween tricks! Visit www.ramseykits.com for the latest pricing, specials, terms and conditions. Thanks Robin... for reminding me of this ad deadline and that Halloween is approaching! Therefore today's color theme is Pantone 021C Orange! *Copyright 2010 Ramsey Electronics, LLC...so there!*

worldmags

Q&A

WHAT'S UP:

Join us as we delve into the basics of electronics as applied to every day problems, like:

- ✓ **High Voltage Sine Wave**
- ✓ **Obscure Battery**
- ✓ **Black Body Heat Source**

■ WITH RUSSELL KINCAID

In this column, I answer questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. **Send all questions and comments to: Q&A@nutsvolts.com**

LARGE CLOCK

Q I have read with real interest your articles on the large clock using the PIC. I would like to make a clock that can be programmed with the Arduino and use non surface-mount chips. (It is hard to breadboard with SMC.)

But I want the complete clock ... meaning LEDs for all the hours, all the minutes, and all the seconds. The seconds would consist of a circle of 60 bi-color LEDs (green and red) with the green on all of the time and the red coming on each second in rotation.

The minutes would consist of a circle of 60 yellow LEDs, each of which would stay lit for the minute indicated and go off when the next minute comes on. Also, when the minute LED lights up it needs to trigger a cluster of LEDs that can be formed into the minute hand.

The hours would consist of 12 LEDs that stay on for the hour indicated and also trigger a cluster that can be formed into the hour hand.

In addition, there needs to be a means of setting the time. This would then be a digitally driven analog clock with no motors or moving parts!

Any advice or help would be greatly appreciated.

— Jay Simmons

A I am not going to learn a new programming language and use an unfamiliar chip, but I could design a circuit and PICBASIC program for a PIC16F57. You can convert the program to C++ for the Arduino and I am sure you can find a version among the 11 now produced that will be compatible.

Wait a minute! Sixty lines for the seconds, 60 lines for the minutes, 12 lines for the minute hands, and 12 lines for the hour hands: 144 lines. I don't know of a micro with 18 eight-bit ports, so it looks like a hardware solution is easier.

Four 74HC154 (one of 16 decoders) will give the 60 lines for the seconds and another four for the minutes. An hour counter will use another 74HC154 to move the hour hand (see **Figure 1**). I could still use a PIC16F57 to control the 74HC154s but the cost in time far outweighs the cost of a half dozen 30 cent chips.

The seconds are displayed by bi-color LEDs (red and green). When the red LED is turned on by Q1, the green goes off because its voltage drop is higher. Instead of having the minute LED go off, I suggest a circle of 60 yellow LEDs, with every fifth one being red to facilitate reading the time. These LEDs will be on all the time. The minute hand will rotate with 60 positions; the hour hand will have 12 positions. You can have up to 20 LEDs in each hand with the ZVP2106A or, if you want to use more, the NDP6020P will handle up to 2,000.

Here is how the circuit works: The line frequency is divided by 60 to produce one second clocks to IC5 which is a binary counter. This drives IC1, IC2, IC3, and IC4 which represent one of 16 decoders. When the count of IC1 gets to 16, feedback through IC14A resets IC5 to zero, which decodes to one. At the same time, IC15, a synchronous shift register, is clocked which enables IC2 and disables IC1. This process continues until we get to the 16th count in IC4. The output from IC4 through IC17F puts IC15 in parallel entry mode and the clock from IC17D does the parallel entry which enables IC1, and the cycle starts over. The reason for the two inverters, IC17D and IC17E, is that the clock has to happen after the parallel entry mode is set, so the clock is delayed by two inverter time delays. The delay is less than a microsecond so it won't be noticed. The minute circuit is identical to the second's circuit, and the hour circuit differs only in that it simply has to count to 12.

To set the clock, the normal operation is disabled and the one second pulses from IC8B are used to advance the count. Switches A, B, C, and D are momentary pushbuttons. Switch A will advance the seconds in 15 second increments and switch B increments in one second increments; you can set the seconds in less than 19 seconds and the minutes the same way. The hours increment one hour per second so that is fast also.

I have not built or simulated this

worldmags

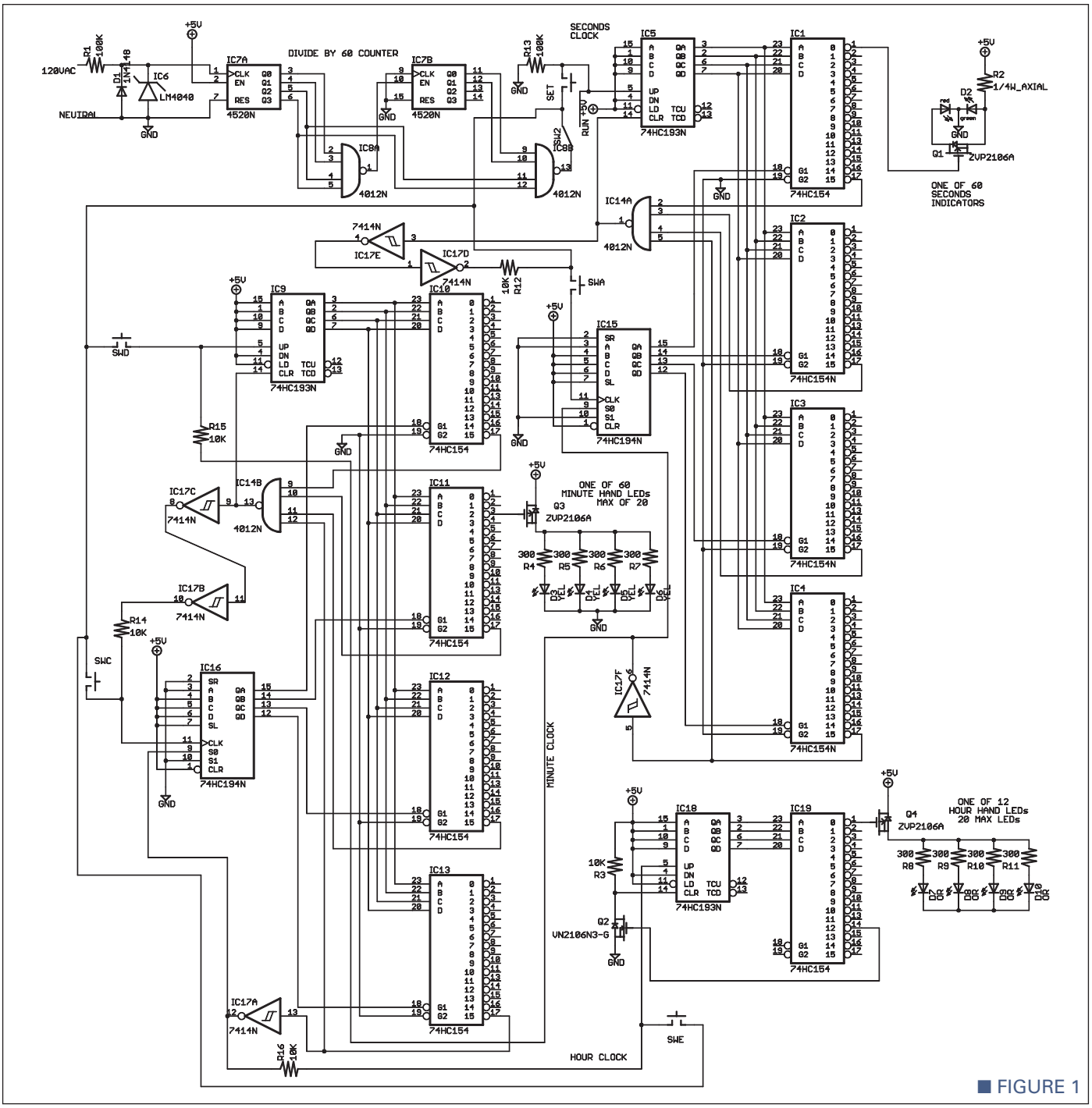


FIGURE 1

circuit, so I will appreciate any feedback or concerns about my logic.

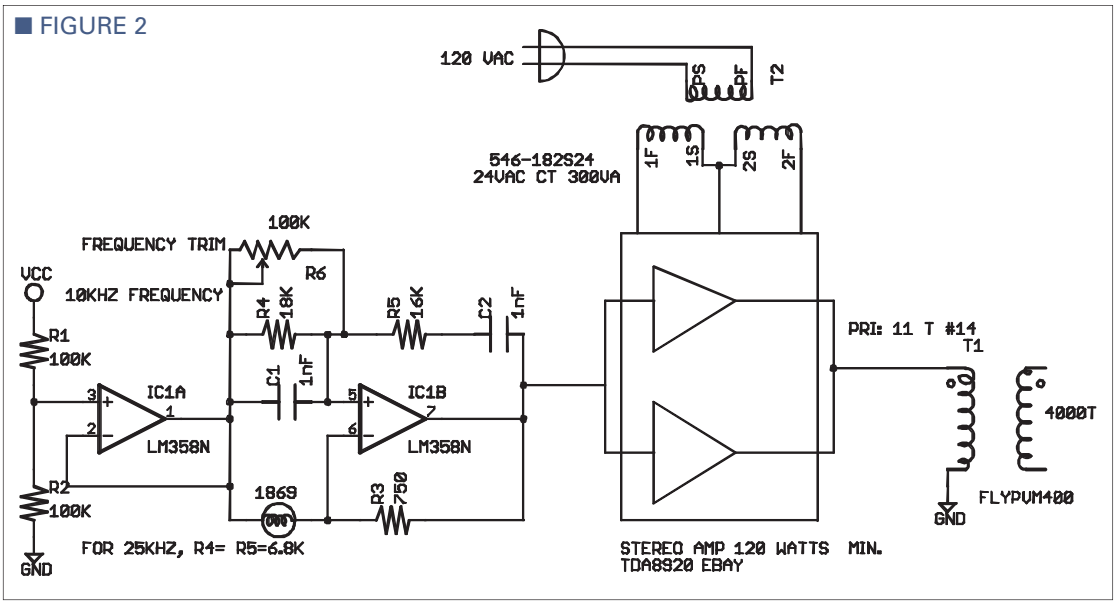
HIGH VOLTAGE SINE WAVE

Q I want to generate a sine signal at 10 kHz and also 10,000 volts for a project that I am working on. I have seen the Royer oscillator at **http://wiki.4kv.org/index.php/royer_oscillator**. This system uses a flyback for the output but I am not sure that it is a sine wave. I also need to make a 25 kHz sine signal at 10,000 volts.

PS: Can you recommend a good electronics dictionary?

A I don't know if the Royer oscillator will be a good sine wave so I am proposing a sine oscillator and linear amplifier (class D) (see **Figure 2**). You don't say what the load is on the 10 KV output, but if you want 10 KV at 10 mA, that is 100 watts. If your power requirements are less, you can reduce the size of the power amp and power transformer. I am going to assume that you use the coil FLYPVM400 from Information Unlimited because I know how many turns are on it. If you use some other

■ FIGURE 2



coil, you will have to experiment to determine the number of primary turns.

In this case, the amplifier is capable of 100 watts into 8 ohms. The voltage output (P=V^2/R) is 28 volts AC. In order to get 10 KV out, you need a turns ratio of 10,000/28 = 357. The primary turns are therefore: 4000/357 = 11. The primary current will be in the order of 357*10 mA = 4 amps, so #14 wire would be good.

I found an amplifier on eBay from China (TDA8920); it's 60 watts stereo and you can parallel the two amplifiers for 120 watts output. The amplifier requires 20-0-20 VAC or 40VCT; Mouser part number 546-182S24 should work. The oscillator circuit is adapted from National Semiconductor Application Note: AN31, Figure 29, Wein bridge oscillator. This is the circuit used by HP in their 200CD oscillator. The 1869 lamp (Mouser part number 606-CM1869) regulates the amplitude because its resistance increases when the signal amplitude increases. The frequency is given by: F = 1/(2*PI*R*C) when R4 = R5 and C1 = C2.

It may be out of print, but I find Rudolf F. Graf's *Modern Dictionary of Electronics*, published by Howard W. Sams to be useful. There is also *Glossary of Communications* by Emerson G. Smith (1971).

OBSCURE BATTERY

Q I am having a problem locating a 12 volt SLA battery for a large toy truck. The battery name is DYNACOLT 6-FM-4.0 12 Volt 2.3 Ah; it has a three terminal Molex connector. I can find some similar batteries but none have the three connections. No one can tell me what the three wire configuration is. I am not sure if a 12 volt two terminal SLA could be configured with the three-pin Molex connector.

— Charles Huth

A I believe the red and black wires are the positive and negative of the battery. The blue wire is probably connected to a thermistor to measure the temperature of the battery. If you charge another battery without the thermistor, you can protect it from overheating by connecting a resistor between the charger and battery. Measure the charger voltage (Vc), then compute the resistor value from: R = (Vc-13.8)/I where I = 2.3 amps divided by the charge time. Ten hours is the standard charge time but you could use less if you check the temperature. The power rating of the resistor is I^2*R.

CIRCUIT SIMULATION SOFTWARE

Q I am trying to find a low cost and easy to use piece of software to simulate circuits. Back in 2008, my high school teacher recommended a program called Circuit Shop and it worked out well for a little more than a year; then I started learning and designing more complex circuits and the software just couldn't keep up. It would return answers on capacitors and inductors that I didn't even have to calculate to know they were wrong. I had a friend just a few months ago recommend LT Spice saying it is what his college uses but warned me that it was less than user friendly. After messing around with it for a little over a month now, I can see what he means. I have to navigate through two or three different windows just to set up the parameters on a resistor. Currently, my college uses MultiSim but that is way out of my price range. Can you recommend some really good circuit simulation programs that are fairly easy to use, reflect what I might see when I get a job in the industry, and won't break the bank?

— Cameron Seidi

A Any circuit simulator is going to involve a learning curve. You need to read the user's manual available at: **http://ltspice.linear.com/software/scad3.pdf**. There is also a user's forum in Yahoo Groups: **http://groups.yahoo.com/group/LTspice**. I have used Ispice, IGspice, Pspice, Electronic Workbench, and am presently using LTspice. LTspice is full functioning freeware and I have not found any program like it. I have Tina-TI from Texas Instruments but

when I tried to use it for a passive circuit, I got an error message: No IC in the circuit. I did a Google search and found 5spice which looks to be easy to use but costs $199. There is a free demo but it is not useful. I also found Opus spice which is free but I have not evaluated it or figured out how to install it.

LTspice has made some modifications that minimize "time step too small" errors in switching circuits so it is good for those types of circuits. In any simulator, the results are only as good as the model, but the more complex the model, the more likely that the program will fail. Always use the simplest model that will give the results you need and keep in mind that if you use an IC in an atypical way, it may not work because of assumptions that the model designer made.

BLACK BODY HEAT SOURCE

Q I need to calibrate some infrared thermometers. I hacked an old black body heat source, modifying the well to accommodate the IR thermometers and an NIST traceable RTD thermometer. My issue is that the source only produces a single (100 deg C) temperature. My goal is to be able to construct a stable multiple point heat source (e.g., 50, 100, 150 deg C) or a variable source of the same range.

Here are some particulars about the old black body heat source, some of which I want to reuse: The heat is provided by a coil of resistive wire which measures 9.5 ohms at 75 deg F, which is wound around a copper bobbin with a well painted flat black. It is powered by a 12V 7 AH SLA battery (nice because it makes it quite portable) and it uses a temperature sensor which measures ~800 ohms at 75 deg F, and uses a uA78S40PC IC to do the control.

The only components that I want to reuse are the well with the heating coil and the 12V power source. Purchased heat sources are mega bucks and I'd like to do this cheaper. Would you please help me?

— Mark Hoffman

A Since the 78S40 has been doing the job, we may as well continue with that. I don't know how much power is required but with 12 volts and 9.5 ohms, it can't be much more than 10 watts. I expect the power to be much less than that if it is well insulated. I will use the step down circuit of the datasheet and calculate the parameters using the Design Formulas from the datasheet.

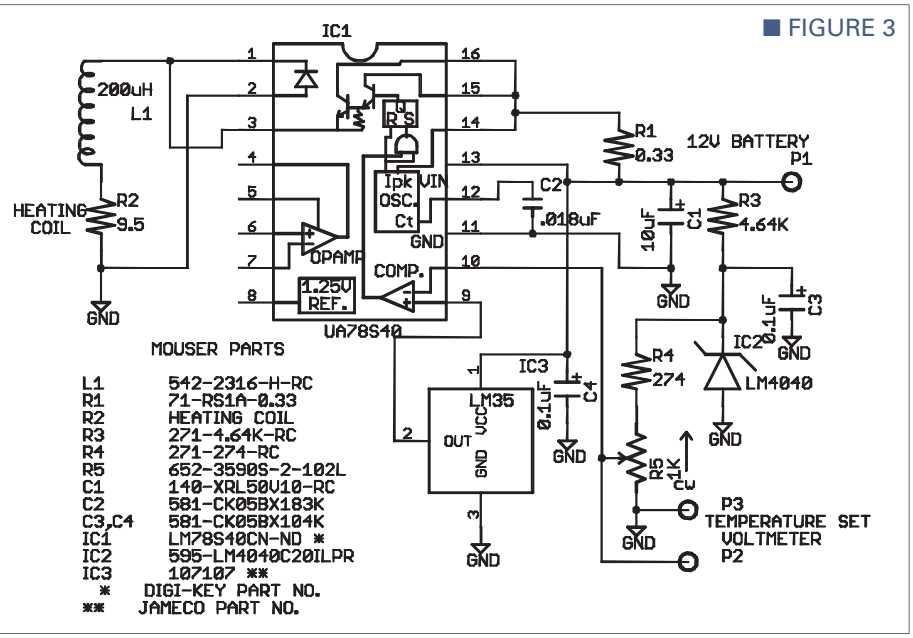
Vo=10V (that's the most we can get with 12V supply)
Vd = 1.5V (diode drop at 1 amp)
Vi=12V (battery)
Vsat=1.3V (transistor drop at 1 amp)

Ton/Toff = 12 (from datasheet formula)

Set Fmin to 20Khz (so you can't hear it)

worldmags

■ FIGURE 3



Ton + Toff = .05 mS = 50uS max = Ton + Ton/12
Ton(1.083)=50uS, Ton = 46 uS

Ct = 4*10-5*46 uS = 18.5 nF (18 or 20 nF is close enough)

Io = 10V/9.5 ohms = 1A = Ipk with no filter cap (we don't need to filter the output)

Ipk = 2*Io max is arbitrary.

Lmin = ((Vi-Vsat-Vo)/Ipk)*Ton = 32uH
I would like the current to be continuous, so won't use the minimum inductance. Simulation indicates that 200 uH will give continuous current at low power (10 uS Ton).

Rsc = 0.33 ohms, 1/2 watt

You will be checking the temperature with your RTD but for temperature feedback, I will use LM35 — a Celsius temperature sensor. The output is 10 mV per deg C, so the output will be 1.5V at 150 deg C. I want to be able to read the set temperature with a digital voltmeter, so I can't use the internal reference of 1.25V. The LM4040-2 (2.048 volt reference) will be able to provide 1.5 volts to set the 150 deg C temperature (see **Figure 3**).

The LM35 should be clamped to the well because the initial temperature overshoot will be proportional to the thermal resistance between the well and the sensor. The TO-46 metal case would be best but I did not find any; the part is plastic (TO-92). I recommend connecting C4 directly to the LM35 pins and use #36 wire to connect to the outside of the insulation to minimize conducted thermal energy. R5 is a 10 turn pot, so you can set the temperature more accurately. **NV**

Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
With our popular Servo Erector Set you can easily build and control the robot of your dreams. Our interchangeable aluminum brackets make the ultimate precision mechanical assemblies. Our Visual Sequencer for SSC-32 provides powerful PC, Atom, or BS2 based control.
Visit our huge website to see our complete line of aluminum and Lexan based robot kits, electronics, and mechanical components.
Biped BRATs
Johnny 5
Round 3DOF Hex
A4WD1
Bipeds
Mini 2DOF Hex
AL5C Arm
www.lynxmotion.com
Toll Free: 866-512-1024
Images represent a fraction of what can be made!
The SES now has 143 unique components!

worldmags

GREAT VALUE IN TEST & MEASUREMENT

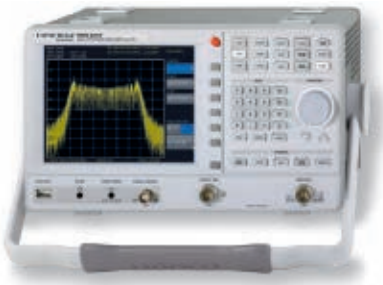
Call for educational discount

350 MHz 2/4 Channel Digital Oscilloscope HMO 3522 / HMO 3524

USB · USB Stick · optional IEEE-488 · optional TCP/IP · DVI Output · inclusive

- ☑ **4 GSa/s Real time, 50 GSa/s Random sampling, low noise flash A/D converter (reference class)**
- ☑ **2 MPts memory per channel, memory Zoom up to 100,000:1**
- ☑ **MSO** (Mixed Signal Opt. HO 3508 / HO 3516) with 8/16 logic channels
- ☑ Vertical sensitivity 1 mV...5 V/div. (into 1 MΩ / 50 Ω) Offset control ± 0.2...± 20 V
- ☑ 12 div. x-axis display range
  20 div. y-axis display range with **V**irtual**S**creen function
- ☑ Trigger modes: slope, video, pulsewidth, logic, delayed, event
- ☑ FFT for spectral analysis
- ☑ Lowest noise fan
- ☑ 6 digit counter, Autoset, automeasurement, formula editor, ratiocursor
- ☑ Crisp 6.5" TFT VGA display, LED backlight, DVI output

3 GHz Spectrum Analyzer HMS 3000 / HMS 3010



1GHz: HMS1000 HMS1010

USB · USB Stick · optional IEEE-488 · optional TCP/IP · DVI Output · inclusive

- ☑ Frequency range 100 kHz...3 GHz
- ☑ Amplitude measurement range -114...+20 dBm
  DANL -135dBm with Preamp. Option HO3011
- ☑ Sweep time 20 ms...1000 s
- ☑ Resolution bandwidth 100 Hz...1 MHz in 1–3 steps, 200 kHz (-3 dB)
  additional 200 Hz, 9 kHz, 120 kHz, 1 MHz (-6 dB)
- ☑ Spectral purity < -100 dBc / Hz (@ 100 kHz)
- ☑ Video bandwidth 10 Hz...1 MHz in 1–3 steps
- ☑ Tracking Generator (HMS 3010) -20 dBm / 0 dBm
- ☑ Integrated AM and FM demodulator (int. speaker)
- ☑ Detectors: Auto-, min-, max-peak, sample, RMS, quasi-peak

Progr. 2/3/4 Channel High-Performance Power Supply HMP Series

USB · RS-232 · optional IEEE-488 · optional TCP/IP · inclusive

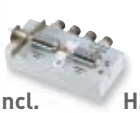
- ☑ HMP2020: 1 x 0...32 V/0...10 A 1 x 0...5.5 V/0...5 A, max. 188 W
- ☑ HMP2030: 2 x 0...32 V/0...5 A 1 x 0...5.5 V/0...5 A, max. 188 W
- ☑ HMP4030: 3 x 0...32 V/0...10 A, max. 384 W
- ☑ HMP4040: 4 x 0...32 V/0...10 A, max. 384 W
- ☑ 188/384W output power realized by intelligent power management
- ☑ Low residual ripple: < 150 μV_{rms} due to linear post regulators
- ☑ High setting- and read-back resolution of up to 1 mV/0.2 mA
- ☑ HMP4030/HMP4040: Keypad for direct parameter entry
- ☑ Galvanically isolated, earth-free and short circuit protected output channels
- ☑ Advanced parallel- and serial operation via V/I tracking
- ☑ EasyArb function for free definable V/I characteristics
- ☑ FuseLink: individual channel combination of electronic fuses
- ☑ Free adjustable overvoltage protection (OVP) for all outputs
- ☑ All parameters clearly displayed via LCD/glowing buttons

25/50 MHz Arbitrary Function Generator HMF2525 / HMF2550

USB · USB Stick · optional IEEE-488 · optional TCP/IP · RS-232 · inclusive

- ☑ Frequency range 10 µHz...25 MHz / 50 MHz
- ☑ Output voltage 5 mV_{pp}...10 V_{pp} (into 50 Ω) DC Offset ±5 mV...5 V
- ☑ Arbitrary waveform generator: 250 MSa/s, 14 Bit, 256 kPts
- ☑ Sine, Square, Pulse, Triangle, Ramp, Arbitrary
  waveforms incl. standard curves (white, pink noise etc.)
- ☑ Total harmonic distortion 0.04 % (f < 100 kHz)
- ☑ Burst, Sweep, Gating, external Trigger
- ☑ Rise time < 8 ns, in pulse mode 8...500 ns variable-edge-time
- ☑ Pulse mode: Frequency range 100 µHz...12.5 MHz / 25 MHz,
  pulse width 10 ns...999 s, resolution 5 ns
- ☑ Modulation modes AM, FM, PM, PWM, FSK (int. and ext.)
- ☑ 10 MHz Timebase: ± 1ppm TCXO, rear I/O BNC connector
- ☑ Front USB connector: save & recall of set-ups and waveforms
- ☑ 3.5" TFT: crisp representation of the waveform and all parameters

LCR - Bridge HM8118



incl. HZ188

USB · RS-232 · optional IEEE-488

- ☑ Basic Accuracy 0.05 %
- ☑ Measurement functions L, C, R, |Z|, X, |Y|, G, B, D, Θ, Δ, D, M, N
- ☑ Test frequencies 20 Hz...200 kHz
- ☑ Up to 12 measurements per second
- ☑ Parallel and Series Mode
- ☑ Binning Interface HO118 (optional) for automatic sorting of components
- ☑ Internal programmable voltage and current bias
- ☑ Transformer parameter measurement
- ☑ External capacitor bias up to 40 V
- ☑ Kelvin cable and 4 wire SMD Test adapter included in delivery
- ☑ Galvanically isolated USB/RS-232 Interface, optional IEEE-488

1,2 GHz/3 GHz RF-Synthesizer HM8134-3/HM 8135

USB · RS-232 · optional IEEE-488

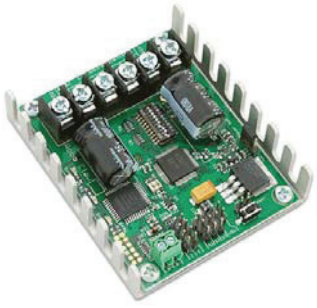
- ☑ Outstanding Frequency range 1 Hz...1,2 GHz / 3 GHz
- ☑ Output power -127...+13 dBm / -135...+13 dBm
- ☑ Frequency resolution 1 Hz (accuracy 0.5 ppm)
- ☑ Input for external time base (10 MHz)
- ☑ Modulation modes: AM, FM, Pulse, Φ, FSK, PSK
- ☑ Rapid pulse modulation: typ. 200 ns
- ☑ Internal modulator (sine, square, triangle, sawtooth) 10 Hz...150 kHz/200 kHz
- ☑ High spectral purity
- ☑ Standard: TCXO (temperature stability: $\pm 0.5 \times 10^{-6}$)
  Optional: OCXO (temperature stability: $\pm 1 \times 10^{-8}$)
- ☑ Galvanically isolated USB/RS-232 Interface, optional IEEE-488
- ☑ 10 configuration memories including turn-on configuration

HAPRO Electronics
Tel: +1-516-794-4080 · Fax: +1-516-794-1855 · sales@haproelectronics.com

www.hameg.com

HAMEG Instruments
A Rohde & Schwarz Company

worldmags



Advance Motor Control

RoboClaw 2X25Amp:
- Quadrature Encoder Support
- Regenerative Breaking
- High Speed Direction Change
- 5V BEC Built In
- Battery Level Monitoring
- Hardware Optical Decoder
- Thermal Protection
- Serial, R/C or Analog Control
- Easy to Use

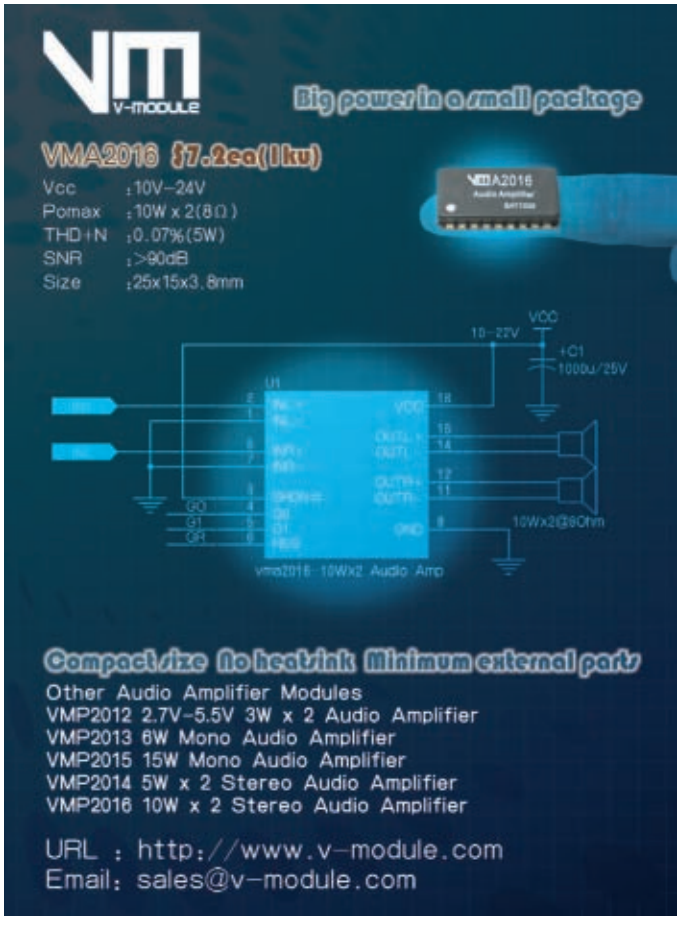
Starting at $59.95

Robot Brains

ARC32:
- Robotics controller
- Built in 32 servo controller
- SSC32 compatible firmware
- SPI, I2C, 2 UARTS, 16 A/D
- Program in C, BASIC or ASM
- Control from a PC USB
- Extensive code libraries

Only $99.95

BASIC MICRO
TECHNOLOGY AT WORK
www.basicmicro.com
(800) 535-9161



** Quality Quick Turn Sheet Metal Products & Precision Machined Parts*

** No Minimum Quantity Restrictions*

** Quote in 24 Hours or Less!*

** Several Metallic & Non-Metallic Materials in Stock*

Visit www.iitmetalfab.com today for your free, no obligation quote!

SIGHT & SOUND KITS FOR ELECTRONIC ENTHUSIASTS

GUITAR KITS

4 Channel Versatile Mixer Kit
KC-5448 $57.50 plus postage & packing

This is an improved version of our popular guitar mixer kit and has a number of enhancements that make it even more versatile. The input sensitivity of each of the four channels is adjustable from a few millivolts to over 1V, so you plug in a range of input signals from a microphone to a line level signal from a CD player etc. A headphone amplifier circuit is included for monitoring purposes. A three stage EQ makes this is a very versatile mixer that will operate from 12VDC, 45mA. Kit includes case, PCB with overlay and all electronic components.

Short Circuits III Kit - Guitar Practice Amp
KJ-8092 $10.00 plus postage & packing

This low cost amplifier allows you to practice your guitar without annoying the neighbors. It has a guitar type input socket, small speaker and volume.

- Kit supplied with PCB, speaker, socket and electronic components.
- 12VDC power required.
- PCB: 105 x 60mm

Instructions NOT included. See KJ-8093 $1.25 for individual instructions or full color project book BJ-8505 $7.50

Short Circuits III Kit - Guitar Link
KJ-8090 $8.75 plus postage & packing

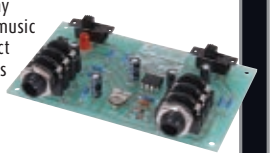
Play the guitar without the need of a guitar amplifier and speakers. This neat pre-amplifier allows you to connect to virtually any amplifier including your home hi-fi.

- Kit includes PCB, 6.5mm sockets and electronic components.
- 12VDC power required.
- PCB: 105 x 60mm

Instructions NOT included.
See KJ-8091 $1.25 for individual instructions or full color project book BJ-8505 $7.50

Short Circuits III Kit - Guitar Distortion Unit
KJ-8102 $10.00 plus postage & packing

Distortion is popular to many guitarists. It can make the music sound quite different. Project includes PCB, 6.5mm sockets and electrical components. 12VDC power required.



PCB: 105 x 60mm

Instructions NOT included. See KJ-8103 $1.25 for individual instructions or full color project book BJ-8505 $7.50

STEREO HEADPHONE DISTRIBUTION AMPLIFIER

KC-5417 $20.50 plus postage & packing

Enables you to drive up to two stereo headphones from any line level (1volt peak to peak) input. The circuit features a facility to drive headphones with impedances from about 8-600 ohms. The Jaycar kit comes with all specified board components and quality fibreglass tinned PCB.

- Power Supply to Suit: KC-5418 $11.75
- PC board size: 134 x 103mm

"THE CHAMP" AUDIO AMPLIFIER

KC-5152 $4.75 plus postage & packing

This tiny module uses the LM386 audio IC, and will deliver 0.5W into 8 ohms from a 9 volt supply making it ideal for all those basic audio projects. It features variable gain, will happily run from 4-12VDC and is smaller than a 9 volt battery, allowing it to fit into the tightest of spaces.

- PCB and all electronic components included.
- PCB: 46 x 26mm

"MINIVOX" VOICE OPERATED RELAY

KC-5172 $9.50 plus postage & packing

Voice operated relays are used for 'hands free' radio communications and some PA applications etc. Instead of pushing a button, this device is activated by the sound of a voice. This tiny kit fits in the tightest spaces and has almost no turn-on delay. 12VDC @ 35mA required. Kit is supplied with PCB electret mic, and all specified components.

PCB: 47 x 44mm

45 SECOND VOICE RECORDER MODULE

KC-5454 $25.50 plus postage & packing

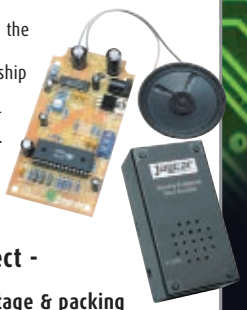
This kit has been improved and can now be set up easily to record two, four or eight different messages for random-access playback or a single message for 'tape mode' playback. Also, it now provides cleaner and glitch-free line-level audio output suitable for feeding an amplifier or PA system. It can be powered from any source of 9-14V DC. Supplied with silk screened and solder masked PCB and all electronic components.

PCB: 120 x 58mm

SCI-FI

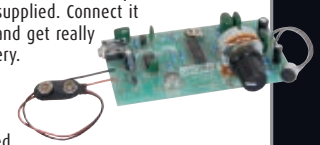
Starship Enterprise Door Sound Emulator
KC-5423 $23.25 plus postage & packing

FOR ALL YOU TREKKIE FANS!
This easy to build kit emulates the unique sound of a cabin door opening or closing on the Starship Enterprise. The sound can be triggered by switch contacts or even fitted to automatic doors. Comes with PCB with overlay, speaker, case and all specified components.
9-12VDC regulated.



Short Circuits II Project - Sound Like a Dalek
KJ-8228 $14.00 plus postage & packing

This project will change your voice into that metallic sounding robot voice so familiar in the space adventure movies. PCB and parts supplied. Connect it to the no brainer amp and get really scary! Requires 9V battery.



PCB: 102 x 57mm

Instructions NOT included.
See KJ-8229 $1.25 for individual instructions or full color project book BJ-8504 $6.50

Short Circuits II Project - Knight Rider Light Scanner
KJ-8236 $11.75 plus postage & packing

Now you can make the light scanner that goes on the bonnet of that famous car. Looks fantastic in all sorts of places! PCB and all parts supplied. Requires 9V battery.

PCB: 102 x 55mm

Instructions NOT included. See KJ-8237 $1.25 for individual instructions or full color project book BJ-8504 $6.50

LED Sabre with Light & Sound Effects
GT-3520 $10.00 plus postage & packing

Challenge your friends in an intergalactic battle and bring balance to the force. Extremely light and durable, made of tough acrylic to take all the knocks of traveling around the galaxy at light speed. Sound and light effects.

Size: 710(L) x 48(Dia)mm

NEW KIT OF THE MONTH

Remote Control Digital Timer Kit
KC-5496 $20.00 plus postage & packing

Remote-controlled digital timer with a bright 20mm-high 7-segment red LED display. It can count up or down from one second to 100 hours in 1-second increments. Its timing period can either be set and controlled using the remote control or it can be automatically controlled via external trigger/reset inputs. An internal relay and buzzer activate when the unit times out. The relay contacts can be used to switch devices rated up to 30VDC or 24VAC and the project can be powered from a plugpack or a battery. Short form kit only - you'll need to add your own universal remote, power supply and enclosure.

- 9-12VDC @300mA
- PCB and components

NEW TO JAYCAR

POST & PACKING CHARGES

Order Value	Cost
$25 - $49.99	$7.50
$50 - $99.99	$20
$100 - $199.99	$40
$200 - $499.99	$60
$500+	$75

Max weight 12lb (5kg).
Heavier parcels POA.
Minimum order $25.

Note: Products are dispatched from Australia, so local customs duty & taxes may apply.
Prices valid until 31/10/2010

HOW TO ORDER

WEB: www.jaycar.com
PHONE: 1-800-784-0263*
FAX: +61 2 8832 3118*
EMAIL: techstore@jaycar.com
POST: P.O. Box 107, Rydalmere NSW 2116 Australia

- ALL PRICING IN US DOLLARS
- MINIMUM ORDER ONLY $25

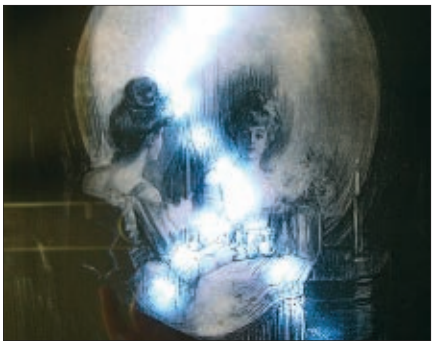
*Australian Eastern Standard Time
(Monday - Friday 09.00 to 17.30 GMT + 10 hours)
Note: 10-14 days for air parcel delivery

Order online: www.jaycar.com

Jaycar Electronics

worldmags

BUILD THE SHAZAM!!



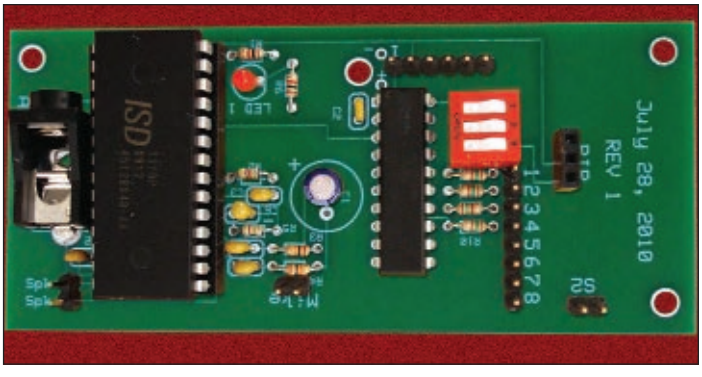
BY RON NEWTON

It was a dark and stormy night. Two small children were walking down the dimly lit sidewalk. The full harvest moon was peeking out every now and then through the small breaks in the clouds. A dog was howling on an eerie note. Spooks, goblins, and vampires could be seen on the other side of the street. The children slowly walked up to door with bags in their small hands. As they crept up closer, on the wall there was a strange picture of a skull or was it a lady sitting at her mirror? As they draw nearer, all of a sudden there is a great flash of light and Shazam! A lightning bolt from the picture flashes with a great clap of thunder!

Halloween is one of America's (and my) favorite holidays. It has become progressively more and more popular with parties, and probably is in second for outdoor displays next to Christmas.

This project was designed to be flexible and reusable for other holidays, and the basic unit can be put together for less than $15 plus the board. No surface soldering is required, so this is an easy first time project. Depending on what type of display you use, the price can increase. Plain LEDs are cheap. I used high output white LEDs which sell for $.89 ea. If you want 100 lumens per LED, the price can jump to $8 ea, and you will need to add drivers. For those who want lighting for a stage production, simply can use photo triacs (Sony S211602F) and drive up to 16 amps 220 volts for each bulb.

If you don't want sound, leave out the sound chip and save money. The board can be triggered using a momentary switch. The one described in this article uses a Parallax Passive Infrared detector (PIR) for detecting the presence of a person. The PIR is also available at RadioShack for $9.99. My project ended up costing about $40. This doesn't have to be a one-time project. I have put in programming pads, so if you have a PIC 2 programmer you can program the light output for Christmas, Valentine's Day, birthdays, etc. Changing the DIP switch will allow you to code many different programs. The voice chip can be reprogrammed by the DIP switch giving you many choices of different tunes. If you use the PIR, it can be utilized in a lot of other projects as it is a plug-in module.



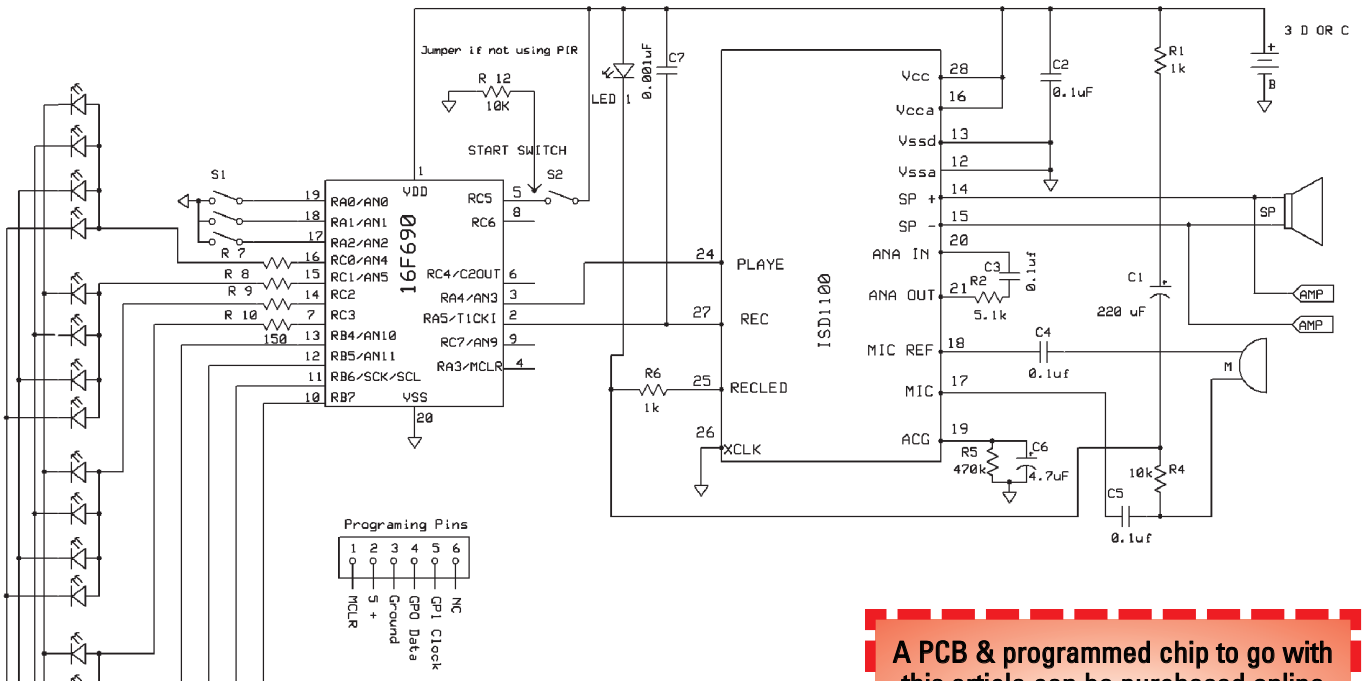
■ FIGURE 1. The printed circuit board.

Display Chip

The heart of the board is a Microchip PIC16F690. It also controls the sound module. The board will provide 30 milliamps for each of the 16 LEDs using a multiplexing technique. The eight outputs are divided into two sections of four outputs ($4^2 = 16$). The LEDs are set up in four groups of four with their anodes tied together. The first cathode of each group is daisychained together. Same with the second, third, and fourth LED. By applying a positive voltage to group one and grounding LED one, it will light; grounding LED two, it will light; and so on.

The PIC is triggered by shorting the two terminals labeled S2. There is also a jack to plug in the PIR which will detect a person's presence using infrared sensing. Both S2 and the PIR provide a positive voltage to trigger the PIC. I added a three-position DIP switch so that you can program nine different modes ($3^2 = 9$). The switch also controls if the voice chip is being used for recording or playing. If you're not using the PIR and if S2 is kept shorted (using a shorting bar), the unit can cycle at 30 seconds, one minute, two minute, and four minute cycles,

worldmags



■ FIGURE 2. Schematic for the frequency synthesizer.

A PCB & programmed chip to go with this article can be purchased online from the *Nuts & Volts* Webstore at www.nutsvolts.com or call our order desk at 800-783-4624.

depending on the DIP switch.

Voice Chip

The voice chip is a CMOS device which uses five volts to power it. It has 10 seconds of memory for storing the sounds, a microphone preamplifier, and speaker outputs. I added a 1/8" jack so that you can plug it into a stereo amplifier for big claps of thunder. Instead of using pushbutton switches as shown in the ISD1110P **schematic** which can be downloaded from the *Nuts & Volts* website (**www.nutsvolts.com**) or from **Jameco.com**. It is driven by the microprocessor so that the timing coincides with the lightning flash.

I got the thunder sound off of the Internet. The "PLAYE" input was used as this allows the PIC to return to its high state and the voice chip will continue to run. The chip is capable of providing 12.2 milliwatts. Once it finishes, it goes to sleep.

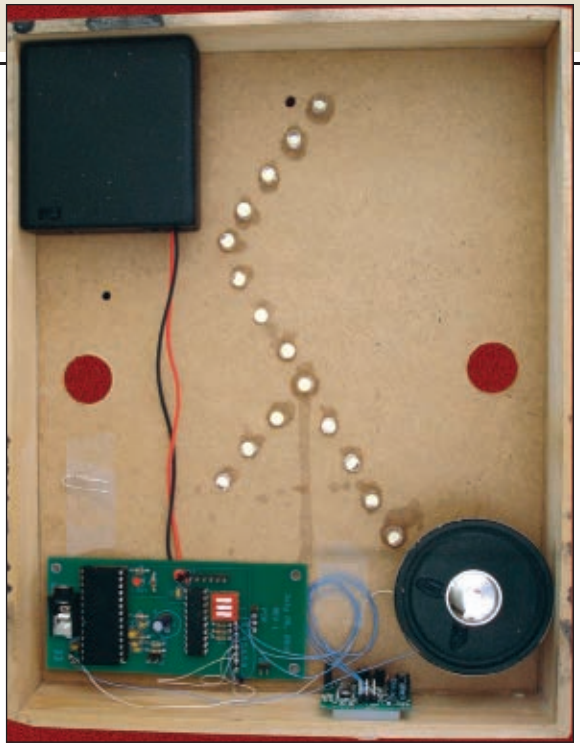
■ FIGURE 3. "All is Vanity" photo.

■ FIGURE 4. The flash.

Construction

The board files, along with several other downloads,

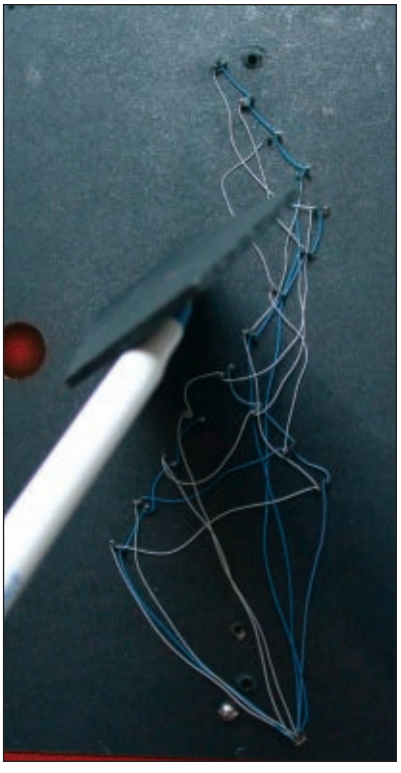
worldmags



■ FIGURE 5. The inside of the frame.

■ FIGURE 6. Inside the frame showing the flash.

are available on the *Nuts & Volts* website (click "Magazine" and then "Downloads" and find "Shazam"). You will need to download the free software from **www.expresspcb.com** to view the schematic and the PCB (printed circuit board). (However, I think you will find it cheaper to order the boards from the *Nuts & Volts* store.)



■ FIGURE 7. The back of the frame.

Also available is a pre-programmed PIC16F690 if you don't want to program it yourself.

If you don't have a PIC 2 programmer and still want to program, put a 20-pin socket in IC1 and solder. Note that the square pin is pin 1 on both ICs. Solder IC2, the resistors, and capacitors into their proper areas. I put an extra pad in for C1 so that you can use either 2.5 mm or 3 mm radial capacitors. Note the "+-" on the 220 µF and 4.7 µF caps. Solder in the three male headers at SPK, the mic, S2, the five male headers in the programming pads, and the three-position female headers in the PIR area. If using a three D, three C, or three AA battery pack, thread the wire through the strain relief hole and solder the red lead to the + and the black to the -. (All chips will run on 4.5 volts.) You can also use a five volt battery eliminator. **Do not use four batteries as the PIR will lock up!**

If programming with a PIC 2, make sure you place the DIP switches in the open position. The chip can also be programmed by other programmers if it's removed from the board. The assembly files are also on the *N&V* website.

Display

The display I built is rather unique. I made a transparency of the picture drawn by Charles Allan Gilbert called "All is Vanity." It is a picture of a woman sitting in front of her mirror and vanity. However, when viewed farther back, it becomes a skull. The picture is available on the *N&V* website if you want to use it.

The frame I used was the cheapest 8" x 11" stand-up frame I could find. The back was removed. I tacked the transparency on each corner to the glass using super glue. A sheet of white tissue paper was tacked to the transparency on each corner giving a semi transparent background but still showing off the picture. I took the back and added a 3/8" x 1" pine strip to make it look like a shadow box mount and painted the outside with a flat black paint. The inside was left unpainted. Holes were drilled in a zig-zag pattern with a #60 drill for the LEDs. The Superbrite LEDs were pushed through the back and held with wire wrap. The circuit board was mounted on the inside of the box in the left lower corner using 1/4" standoffs and 6-32 screws. However, I found that the PIR would *not* detect through either the transparency or the glass. I ended up drilling a 7/8" hole through the bottom of the picture frame and inserted the PIR into this hole from the inside, then tacked it with hot glue. I used a 3-pin male header and plugged it into the PIR female header, then wire-wrapped the PIR using the wire as an extension.

There are mounting holes in the board, and the 1/2"

worldmags

6-32 screws will self tap. When the PIC is triggered, a bright flash of lightning comes from the picture. The frame was placed above my stereo speakers and the output from the voice chip was fed into the stereo input. A clap of thunder comes from the stereo speakers and really makes people jump.

If you don't want to use this particular picture, you can also mount the LEDs on a piece of white (how about a ghost hanging from the ceiling) or black felt (in a dark corner), depending on the effect you want. Zig-zag the LEDs in groups of four. There are a hundred and one possibilities!

Starting from the top of the zig-zag, use a wire-wrapping tool and 30 gauge wire wrap (available from RadioShack). Wrap the anodes of the LEDs in sets of four, e.g., LEDs 1, 2, 3 and 4 to pin one. Then do 5, 6, 7, and 8 to pin 2; 9, 10, 11, and 12 to pin 3; and finally tie 13, 14, 15, and 16 to pin 4.

Daisychain the first LED's cathode of each set together and wire to pin 5, e.g., LEDs 1, 5, 9,and 13 to pin 5; LEDs 2, 6, 10, and 14 to pin 6; LEDs 3, 7, 11, and 15 to pin 7; and the last LEDs 4, 8, 12, and 16 to pin 7. It is important to keep the sets in order for the flash to appear going downwards (see **Figure 7**).

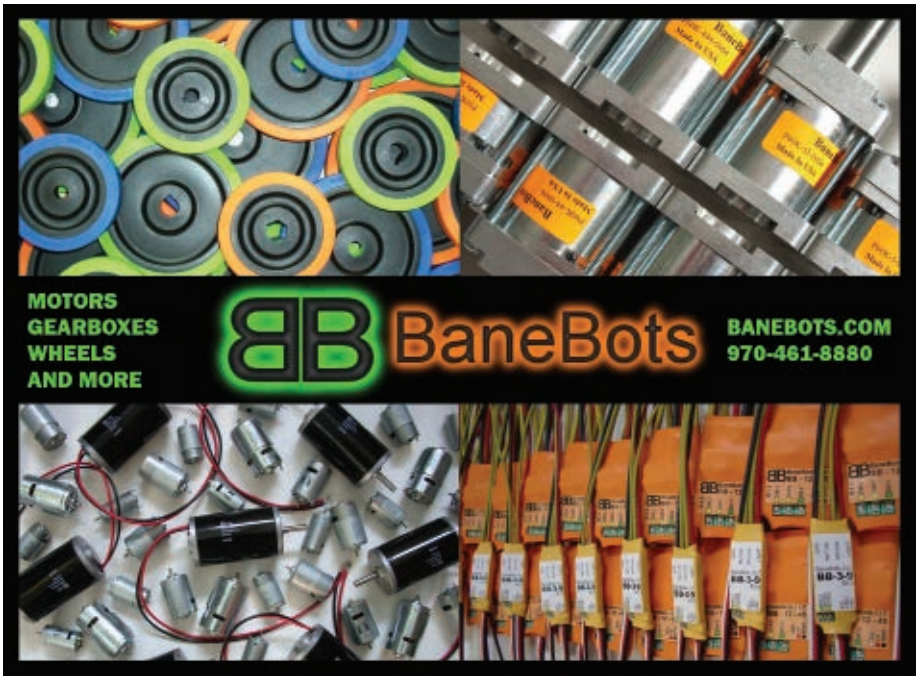
PIR

The PIR sensor is a pyroelectric device that detects motion by measuring change in the infrared (heat) levels emitted by surrounding objects. It uses a crystalline material that generates an electric charge when exposed to infrared radiation. If used, it takes about 60 seconds before it functions properly. The jumper should be placed on L. The PIR specifications state the voltage for the chip should be from three to five volts. Although it would fire, it would not activate the PIC using six volts.

Software

I placed the reading for the trigger at the beginning of the program so that you can change the DIP switch any time. After it is triggered, it will read the DIP switches.

Some of you who are new to programming may wonder what the "ANDWF" command does in the "STARTING AREA." The DIP switches for programming the chip are connected to port A's RA0, RA1, and RA2. These pins are configured to have pulled-up resistors, so all we have to do is to change the DIP switches from open to closed to configure the pins either high or low. There are other connections to port A besides the DIP switch. To read just the first three pins, you use an AND command. You can AND B'00000111' (B is for binary) or you could also use a ".7" decimal number. A zero will eliminate any ones. The ones allow either a one or zero to be read. If we just AND port



DimensionEngineering
The New King of the Jungle
Sabertooth
2x50HV
- Dual Channel Brushed
Motor Controller
- 12V - 48V Nominal / 62V MAX
- 50A Continuous / 100A Peak x 2
- 4.17" x 3.5" x 1.85" / 11oz/292g
- Analog, R/C, and Serial Inputs
www.dimensionengineering.com/Sabertooth2x50HV.htm

worldmags

PARTS LIST

ITEM	DESCRIPTION	QTY	SOURCE
C1	220 µF 6.3 V	1 ea	
C2	.001 µF	1 ea	
C3 – C5 – C7	.1 µF	3 ea	
C6	4.7 µF 16 V	1 ea	
Header female	3 position .1"	1 ea	
Header male	2 position .1"	3 ea	
Header male	6 position .1"	1 ea	
Header male	8 position .1"	1 ea	
IC1	PIC16F690	1 ea	Microchip.com
IC2	ISD1110P	1 ea	Jameco.com
Jack	1/8" mono	1 ea	
LED	3 mm red	1 ea	
R1-R6	1K 1/6W	2 ea	
R2	5.1K 1/6W	1 ea	
R3-R4-R11-R12*	10K 1/6W	4 ea	
R5	470K 1/6W	1 ea	
R7-R8-R9-R10	150 1/6W	4 ea	
S1	Three-position DIP	1 ea	
*R12 jumper if not using PIR			
MISC			
LEDs 1-16	White 18000	16 ea	SuperBrightLeds.com
Motion detector	PIR Sensor	1 ea	Parallax.com
Wire-wrap wire	30 gauge		
Battery holder	C or D (four battery)	2 ea	

A, it will change its outputs which you don't want it to do. So, you place port A into a TEMP register and AND the TEMP registers instead.

After "START," the strobe effect is created by using a series of flags and setting the flags after an LED is turned on. The first time it is turned on, it jumps back to S1. However, once turned on, its flag is set and it will continue on to the next LED. Note that all the flags are cleared at the beginning of the "START" sequence.

Once the strobing is finished, it turns on the voice chip and continues with a series of delays and flashes. The "CALL" feature is a great way of performing the same function over and over. You simply call the procedure you want; at the end of the procedure, you place a return and it will go back to the command after the CALL."

Running Shazam

Use either a small 16 ohm speaker or (better yet) plug your stereo into the jack. Set all the DIP switches to open. Put in the batteries. Place R12 across the PIR pins 1 and 3; short S2. The LEDs should flash downward and then the voice chip will activate. The LEDs will flash again. If you leave S2 shorted, it will flash every 30 seconds. You can change the DIP switches to increase the time. If you short and release, it will flash each time you short after the sound finishes at the time level the DIP switches have been set to. R12 prevents false triggering when using a switch.

Remove R12 and plug the PIR into the female headers on the Shazam board (if you are using it). When it detects the presence of a person, it will flash and wait for 30 seconds before flashing again. Delays can be set via the DIP switches so it won't keep flashing.

Recording

Remove the PIR module. Connect a microphone or tie the headers into the earphone outlet of your computer and connect it to the headers titled "MIC." Put one DIP switch in the closed position and the other two in the open position. When ready, short S2, open S2, and short again to record. The LED will light indicating that the chip is recording. Wait until the light turns off.

There are many thunder sounds available on the Internet. Just perform a search for "thunder sounds." Use a 1/8" plug with about 6" of wire-wrap; put the plug into the ear phone jack of your computer and wrap the wire around the "Mic" pins. Push both the play button on your computer and short S2 at the same time to record. The chip will record 10 seconds of sound. If you goof, try again.

Happy spooking!! **NV**

worldmags

THE ORIGINAL SINCE 1994
PCB-POOL®
Beta LAYOUT
Servicing your complete PCB prototype needs :
• Low Cost - High Quality
PCB Prototypes
• Easy online Ordering
• Full DRC included
• Lead-times from 24 hrs
NEW !
• Optional Chemical Tin finish
no extra cost
NEW !
FREE LASER STENCIL WITH ALL PROTOTYPE PCB-ORDERS
Watch "ur" PCB®
Follow the production of your PCB in
REALTIME
email : sales@pcb-pool.com
Toll Free USA : 1 877 390 8541
www.pcb-pool.com
Beta LAYOUT

FRONT PANELS & ENCLOSURES
Customized front panels can be easily designed with our free software Front Panel Designer
• Cost-effective prototypes and production runs
• Wide range of materials or customization of provided material
• Automatic price calculation
• Fabrication in 1, 3 or 5 days
Visit us at Electronica 2010, along with Schaeffer AG, in Hall B5, Booth 519
New Munich Trade Fair Centre
09–12 November 2010
electronica 2010
components | systems | applications
FRONT PANEL EXPRESS
www.frontpanelexpress.com
(206) 768-0602

PAiA
Back-to-school discounts on basic kits at paia.com
Electronic Kits for Enthusiasts
BUILD YOUR OWN!
PAiA 9700s
Modular Synthesizer
FatMan
Analog MIDI Synth
Theremax
The Ultimate Theremin
Check out these and other exciting D.I.Y. projects!
PAiA
paia.com
Music/Tutorials/Kits
Guitar Effects
Tube Electronics
Studio Gear

worldmags

MAKE MAGIC CANDLES WITH THE PROPELLER

By Jon McPhalen

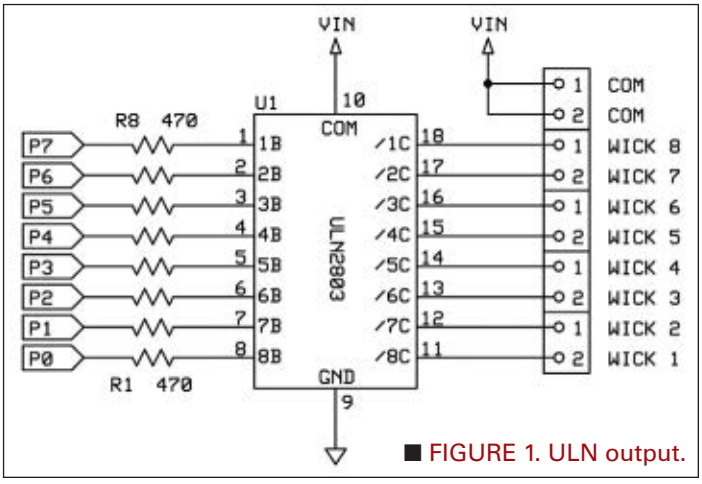
If you're like me, October is a very busy month, mostly trying to finish building props and decorations (some from last year!) for Halloween. Last minute ideas abound, but most just don't fall into the time constraints I usually find myself boxed in by.

A really great way to add atmosphere to a Halloween display is with candles. The real ones, though, can be dangerous, go out if the wind blows, and cannot be controlled with any sort of [practical] automation. There are electronic candles available in retail and specialty stores, but most are not terribly bright and — like their real counterparts — cannot be controlled (without some major hacking).

There is good news. With some simple parts, you can whip up a custom candle controller that will run eight outputs, and can easily be expanded for more. In addition to candle outputs, there is a trigger input to control behavior, and at the end I'll show you how a few more components can be added for additional control; for example, playing audio from an external device.

Building a Candle Controller

My processor of choice is the Parallax Propeller chip. Having worked with it almost daily for the past 18 months, I have a real comfort level and enjoy developing code for it. It's high level language, Spin, is fairly easy to use and the interactive nature of the IDE simplifies and expedites experimenting with new code.



■ FIGURE 1. ULN output.

In order to make this an "afternoon project," I hand-wired the circuit to a Propeller proto board. These boards are an inexpensive way to get started with the Propeller. If you're going to do this as a one-off and never use the Propeller again, you can get a USB version of the proto board. If you think the Propeller is in your future, then you can go with the cheaper (non-USB) version and use a Prop-Clip to program it.

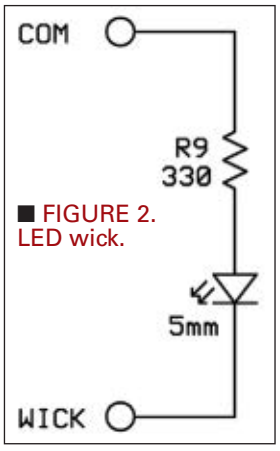
Let's look at the circuitry first. The candle outputs are driven by a ULN2803A; this lets us use anything from simple LEDs to some medium current monsters. With eight outputs running, you should limit the current to about 180 mA through each ULN channel. When it comes to LEDs, that's a LOT of current. **Figure 1** shows the ULN connections to the Propeller. Advanced users will note that resistors R1-R8 are, in fact, optional but having seen ULNs melt internally (usually from an oversized inductive load) and do bad things on the processor side, these resistors are cheap insurance.

There's nothing to the "wick" circuit shown in **Figure 2**. I'm using a 12 volt supply and the resistor shown allows about 20 mA through the LED. The LED, in fact, is the trickiest part of the project. Finding one with the right color and brightness can be a chore. I found that the NTE30039 has the right color (nice warm orange) and is very bright (7,000 mcd).

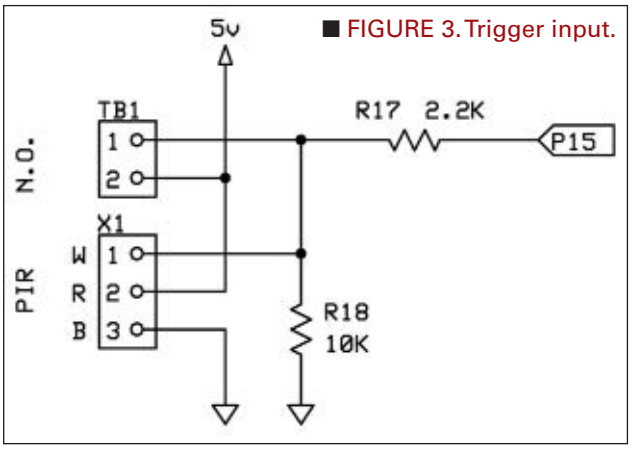
We could stop there and have a lot of fun, but why stop? On Halloween, the ToTs (Trick-or-Treaters) will wander

worldmags

up the sidewalk and it might be fun to have the candles change as they approach. The trigger circuit shown in **Figure 3** accommodates a Parallax PIR sensor (via a three-pin header X1) or a normally open contact (usually a mat switch) connected to TB1. The voltage for the PIR and the normally open contact is 5V, so a 2.2K series resistor is used to limit the current into the Propeller (which operates at 3.3V).



■ FIGURE 2. LED wick.



■ FIGURE 3. Trigger input.

Construction is point-to-point and as it can be difficult to see these connections in a photo, I created the graphic in **Figure 4** as a guide. Note that the resistors between P0-P7 and the ULN2803A are stood up on end. Also note that while it looks like the wires go on top, they are actually soldered to the bottom. Finally, the colored dots on the PIR header show cable orientation (ground is at the top of the board). **Figure 5** shows my completed prototype.

With a few items from the hardware store, we can turn the simple LED circuit into a nice looking flame. The first thing we have to do is modify the LED. I've never been able to find bright LEDs that are diffused; the bright ones tend to come in a water-clear package. This makes the output very beam-like, as the convex end of the LED acts as a lens. We can fix that with a file or a bit of sandpaper. By grinding off the dome at the end of the LED, we will get more light shining from the side of the package. **Figure 6** shows unmodified and modified LEDs.

The next step is to solder the resistor to the LED, add leads, and then protect everything with heat shrink tubing. In addition to protecting the connections, the heat shrink adds a little stiffness that will be useful in the final step. **Figure 7** shows a wick in process and one that is completed. I tend to solder the resistor to the anode side of the LED.

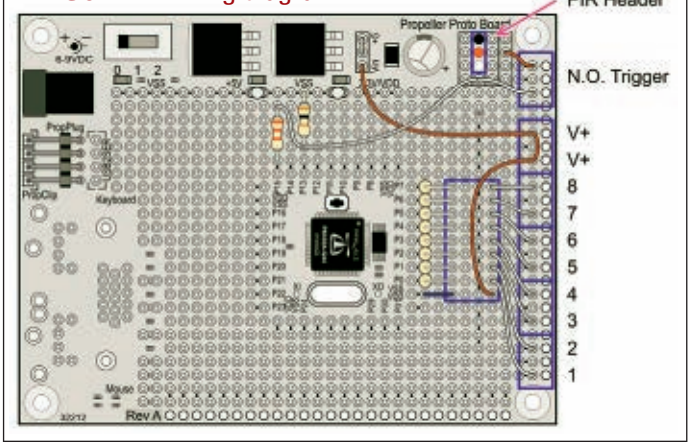
If you want to create a flameless faux candle, you can use the wick as is; simply install it in a candle body, recessed enough to provide the illumination created by a hidden flame. You can actually take real candles (pillar type), burn them down a bit, and then drill a hole through the center of the body. By installing the LED in the wax candle, the light will shine through the translucent paraffin and create a beautiful effect.

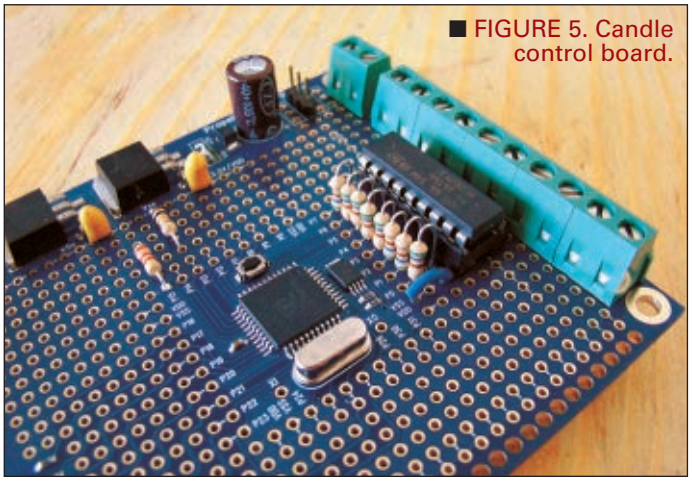
To create a "flame" for the wick, I use clear silicone — the type used to seal seams around bathtubs and sinks. The problem with silicone is that it takes an annoyingly long time to fully cure. This can be fixed by adding two drops of water and two drops of glycerin into two ounces of silicone. In order to ensure a complete mix, I dip the end of a match stick into yellow acrylic paint and then dab that into the silicone. (A tiny bit of color goes a long way.) The idea is not to color the silicone so much to change it, but to provide a visual indication that the water and glycerin are completely mixed in. By adding the water and glycerin, the wicks will be ready in a couple hours versus the 24 normally required for the silicone to cure.

After mixing the silicone in a small cup, dip the LED into it, swirl it around a bit, and then slowly pull it out to create a natural, wispy flame shape. You must keep the LED in the upside-down position until the silicone is completely cured (give it two hours if you mixed in the water and glycerin). **Figure 8** shows a completed wick/flame.

For many projects, I build faux candles from PVC pipe. Simply cut it, clean it, and apply hot glue drips to the top to finish the body. The wick is installed and secured with hot glue; at this point, I have a candle ready for a

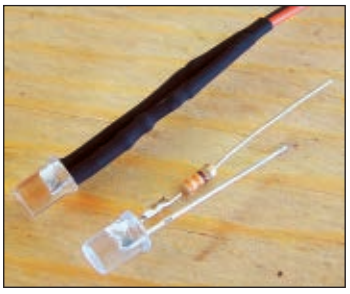
■ FIGURE 4. Wiring diagram.




■ FIGURE 5. Candle control board.

worldmags

■ FIGURE 6. Modified LED.



■ FIGURE 7. LED and resistor.

■ FIGURE 8. Final LED wick.

■ FIGURE 9. Completed candle.

prop or set. **Figure 9** shows a small candle created with my LED wick and a piece of 1/2" PVC pipe.

Programming a Jack-o-Lantern

Even if you're fairly new to using Spin, you should find my demo program easy to modify after I've run you through the explanation. For those that are using the Propeller for the first time, I think you'll find its behavior and programming interesting.

The program is composed of three elements: 1) the main code; 2) a dimmer module to control LED brightness; and 3) a flame effect generator.

Each of these modules runs in their own cog (processor) inside the Propeller so we don't have to worry about managing interrupts (the Propeller doesn't have them) or other time-slicing strategies; trust me, multiple processors on one slab of silicon makes life very pleasant.

When the Propeller boots up, it will load the Spin interpreter into Cog 0 and run our main program. Inside that program, we will instantiate the other cogs. The first is a dimmer cog that runs ***jm_bam8***; this is used to modulate the eight LED outputs using a process called Bit Angle Modulation (for details on how BAM works in the Propeller, see The Spin Zone column in the November '09 issue of *Nuts & Volts*). We don't have to understand how BAM works to use it, just what to do with the object.

The first thing we need to do is include it in the code. This is accomplished by adding it to the objects list in the main program:

```
obj
  leds : "jm_bam8"
```

This includes the file **jm_bam8.spin** in the compilation and creates an object called ***leds***. Most objects that deal with I/O will require some sort of setup; usually with a method called ***start***() or ***init***() — the latter is what I tend to use in my objects.

To instantiate ***leds***, we'll call its ***init***() method with the first pin of an eight-pin group; in my program, there is a constant called WICK1 that is that pin (0 for the hardware we just built):

```
leds.init(WICK1)
```

This line of code launches the BAM processor into Cog 1 and will continue to run until we stop it with the ***finalize***() method. To change the brightness of any of the LED wick outputs, we can use the ***set***() method with a value from 0 (off) to 255 (full brightness). Having just discussed the BAM object and knowing that our program needs a flame generator, you may be wondering why that object is not in the object list above. Let me show you. Here's the main loop that runs my Jack-o-Lantern:

```
pub main | ch, bright, delay

  lottery := cnt

  leds.init(WICK1)

  bytefill(@level, 0, 8)
  cognew(flicker(3), @stack)

  repeat ch from 0 to 2
    repeat bright from 0 to 255
      pause(4)
      level[ch] := leds.ezlog(bright)

  pause(2000)

  repeat
    repeat bright from 255 to 64
      pause(4)
      bytefill(@level,leds.ezlog(bright),3)
      flickerdelay := (255-bright) * 52 / 10

    waitstart(150)

    ?lottery
    delay := (||lottery//8001) + 2000
    pause(delay)

    flickerdelay := 0
    repeat ch from 0 to 2
      level[ch] := 255
      pause(750)

    pause(30000)
```

PARTS LIST

ITEM	DESCRIPTION	SOURCE/PART#
LED1-LED8	Amber, 5 mm	NTE30039
Q1	2N3904, NPN	**Mouser.com** 610-2N3904
R1-R8, R19	470 ohm	Mouser 291-470-RC
R9-R16	330 ohm	Mouser 291-330-RC
R17	2.2K ohm	Mouser 291-2.2K-RC
R18	10K ohm	Mouser 291-10K-RC
S-S1	Socket, 18-pin	Mouser 571-1-390261-5
TB1-TB7	Term block	Mouser 571-2828362
U1	ULN2803A	Mouser 511-ULN2803A
X1, X2, JP1	0.1 M-STRT	Mouser 517-6111TG
JMPR	0.1 shunt	Mouser 517-950-00
Propeller Proto Board		**Parallax.com** 32212 Parallax 32812 (USB)
Hook-up wire		
Heat shrink tubing		
Clear (bathroom) silicone		
Water		
Glycerin		
Acrylic paint (yellow or orange)		

worldmags

In the course of the program, I want to do some random timing so I've created a variable called *lottery* to hold a random number. This variable is initially seeded with the value of the system counter. I like to do this because the Propeller boots using an internal RC clock source which will always have a bit of variance.

We've already discussed starting the BAM driver, so we can move on to getting the flame simulation going. An eight-byte array called *level* will be used to control the maximum brightness of each candle output. As with the BAM driver, the values in *level* are 0 (off) to 255 (maximum brightness).

All of the output levels are set to zero by using ***bytefill***. The next step is to start the flame generator which is, in fact, a method called ***flicker***() that is part of the main code listing. Yes, that's right: We can launch a method into its own cog to run parallel with other processes. The cool thing about using this technique is that the method we launch can still access global variables and other methods in the program.

This is how we launch the ***flicker***() method into its own cog:

```
cognew(flicker(3), @stack)
```

In this particular case, we're telling the flicker generator that we have three wicks (pins 0, 1, and 2); by setting the actual number of wicks used, we have more flexibility in how fast the routine runs.

When using ***cognew*** to launch another Spin cog, we need to provide the name of the method (along with parameters, if any) and the address of a block of longs that will serve as stack space. The stack is used to hold local variables, as well as working variables and return addresses so that other methods can be called.

To be honest, setting the stack size can be something akin to black magic — this is not unique to the Propeller. In the past, I've tended to start with 16 and then add up the number of local variables used in the method. That strategy didn't work in this case and the program just stopped. As soon as I bumped up the stack, everything worked fine. I generally believe it's best to be generous with stack space. If you get concerned about memory usage, you can find tools in the Parallax Object Exchange that will analyze stack usage and let you optimize for the smallest RAM footprint.

Now that all of the elements are running, we can get into the working code. I start by ramping up the brightness of each candle output just so I can see that they're working. With 256 steps and a 4 ms delay between each, it will take about a second for each candle to light to full brightness. This holds for two seconds and then drops into the loop that will run until we power-down.

I often joke that my Halloween prop control programs are like a Quentin Tarantino movie in that they start at the end and then loop back. You can see this at the start of the loop: The brightness of all candles is being dimmed (from 255 to 64) and the flicker speed is being modified to go from 0 (fast, angry) to 1,000 (slow, calm). At the end of this loop, we're back at the idle state with calm, low brightness wicks.

You'll remember that the circuit is set up to accept a Parallax PIR or a simple, normally open switch (usually a mat switch). We're going to use this input to bring Jack to life when a ToT approaches. It's always a good idea to debounce digital inputs and it is especially important with PIRs as they can be twitchy. We can wait for a good input with a call to ***waitstart***():

```
waitstart(150)
```

This call waits for the start input to be active — and stay active — for 150 milliseconds before being considered a valid input. This timing will prevent spurious outputs from the PIR from triggering the prop. The ***waitstart***() method is pretty straightforward:

```
pub waitstart(ms) | debounce

  ms := 10 #> ms <# 1_000

  debounce := 0
  repeat until (debounce => ms)
    pause(1)
    debounce := ++debounce * ina[START]
```

This method "fixes" the input parameter so that the valid range for debouncing an input is 10 milliseconds to a full second. Spin has a numeric range from negative two billion (plus) to positive two billion (plus) — this little correction keeps a typo from hanging up the program.

Here's how the method works: A counter (called *debounce*) is cleared and then we drop into a loop that waits one millisecond before incrementing *debounce* and then multiplying it by the state of the START input pin. If the pin is active (1), then the count will be maintained; if the input is not active (0), then the *debounce* variable will be cleared. This process ensures that the input is active and stays active for the desired period. Once that happens, we return back to the main program.

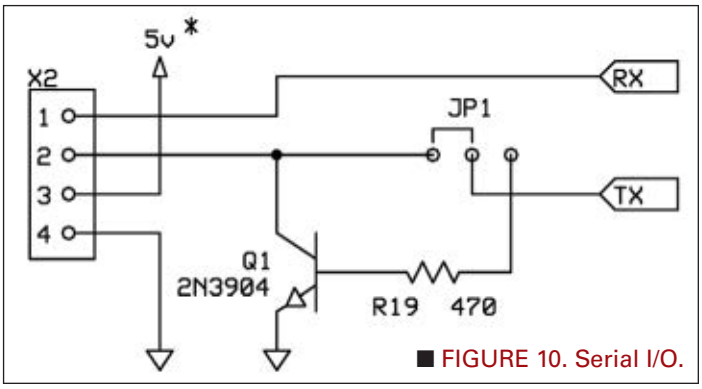
In my props, I like to insert a randomized delay between the trigger event and the active prop code. To do this we randomize the value in *lottery* with ***?*** and then perform a little math to get a value between two and 10 seconds. Note that we have to take the absolute (||) value of *lottery* in the code; this is necessary because a negative value (which is possible with the randomization) will create a problem for the modulus (//) operator.

After the random delay, the flicker rate is set to zero which makes the flames more active and then each is bumped to full brightness with a short delay in between. It's easy and very effective.

Here's why: Imagine ToTs approaching your home where they see a friendly Jack-o-Lantern on the porch, gently lit and looking peaceful. As they draw near, the flames are magically brightened and seem a bit angry. In the minds of the ToTs, you've just created a "What's next?" scenario. They will be [happily] on edge until they're safely back on the sidewalk with the treats you've just provided. The reason for the random delay after the trigger point is to fool returning ToTs.

The candles burn brightly for 30 seconds before going back to the top of the loop (where we started) and are

worldmags



■ FIGURE 10. Serial I/O.

dimmed back down to the idle level.

Again, this is very simple and with a little extra hardware and code you could even add audio control. The circuit in **Figure 10** can be used to start audio players that require an open collector start pulse, e.g., Cowlacious (set the jumper to select the transistor and pulse the TX pin low-high-low), or to communicate with a serial audio player (e.g., EFX-TEK, Rogue Robotics) when the transistor is bypassed. Serial devices will need a serial cog but that's not a problem as the base Jack-o-Lantern code only uses three of the eight available.

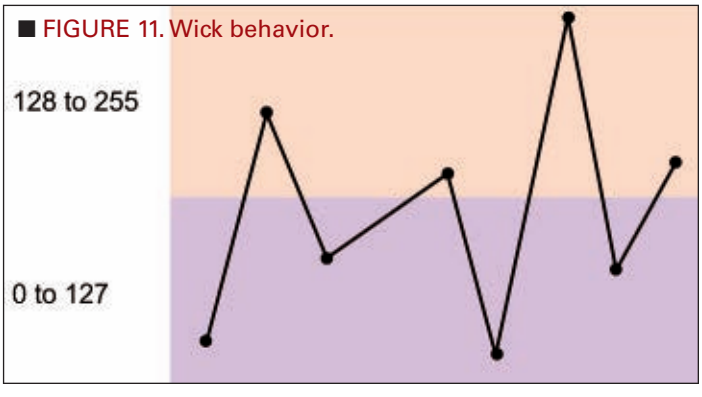
I suspect many will stop right here and head to the workbench to build a candle controller. For those hearty types who want to know how I generate realistic flames, read on!

Mimicking Nature is Difficult

My friend, Jen, was having dinner at a nice restaurant and noticed they were using electronic candles – and they looked awful. I understand; having experimented with candle algorithms, I can tell you it's a little tricky to get a natural look (though it won't be after you read this!).

There is good news: We don't have to do a *perfect* simulation. You see, our brains have this incredible ability to take partial information and fill it out with what we know; this gives us the ability to recognize friends from bad photos at odd angles. So, you see, we only have to get close; our brain will say, "Okay, that's a candle."

Don't believe me? A few years back I had two candles — one real, one electronic — "burning" side by side so I could watch them. The air conditioner kicked on and blew out the real candle. Without thinking, I picked up a match and held it to the electric candle expecting it to light. After about a second, I laughed at myself – out loud! – for being fooled by my own creation. This illustrates the power of our subconscious and its ability to deal with partial (even false) information.



■ FIGURE 11. Wick behavior.

After lots of experiments, I've developed an algorithm that works like this: Each "wick" requires three variables: 1) current brightness; 2) target brightness; and 3) ramp rate moving from the current brightness to the target. When the current level reaches the target level, a new random target is generated. Of late, I am also keeping track of the directional change to ensure that the next update always passes through the mid-point of the brightness range.

Figure 11 is a simplified illustration of the algorithm's behavior (the ramp is actually a bit "wobbly" because the ramp variable is recreated when it hits zero). When the new direction is ramping up, the brightness will be selected in the upper region; when the new direction is ramping down, the brightness is selected in the lower region. This seems to give the wicks a bit more life (as we would want in a display). In the final code, I also apply an overall level control so I can control the maximum brightness.

Coding Candles

Keeping track of up to eight "live" wicks really dictates that it happens as a background process — this allows us to monitor sensors and other things in the main code without affecting wick behavior (unless we want to). As discussed earlier, the process is coded into a Spin method which launched into its own cog:

```
pri flicker(wix) | lotto, idx, tmp, direction {
                 } base[8], target[8], ramp[8]

  lotto := cnt
  repeat idx from 0 to 7
    target[idx] := ?lotto & $FF | $80
    base[idx] := 0
    ramp[idx] := (?lotto & %11) + 1
  direction := %00000000

  repeat ' flicker loop
    ?lotto
    repeat idx from 0 to (wix-1)
      if (—ramp[idx] == 0)
        ramp[idx] := (?lottery & %11) + 1
        if (base[idx] < target[idx])
          ++base[idx]
        elseif (base[idx] > target[idx])

          —base[idx]

        else

          direction := togglebit(direction,
          idx)
          if (getbit(direction, idx) == 0)
            target[idx] := ?lotto & $FF | $80
          else
            target[idx] := ?lotto & $7F

        tmp := (base[idx] * level[idx]) / 255
        leds.set(idx, tmp)

    if (flickerdelay > 0)
      tmp := (flickerdelay * US_001) #> 400
      waitcnt(tmp + cnt)
```

Okay, it looks a little scary, right? Hey, this is the Halloween issue! But just like your friend behind the mask,

worldmags

once you see what's underneath you'll see that it is not as scary as it seems at first blush.

There are two loops: the first sets up initial values; the second runs the active flicker algorithm. Most of the variables used by the ***flicker***() method are local, but you'll see that it uses the external *level* array and the ***leds*** object to control the wicks without further intervention by us. The code also accesses the (global) *flickerdelay* variable which affects the speed of wick movement.

At the top, a local random variable (*lotto*) is initialized and a loop runs which sets the base value of each wick to zero — the target in the upper range — and then randomizes the ramp control. The *direction* bits are cleared to zero which indicates that each wick is initially ramping up.

Now for the fun. At the start of each loop iteration, the value in *lotto* is re-randomized so that it's always changing. For each wick, we decrement the ramp value and when that hits zero we increment or decrement the base value so that it moves toward the target. Once the target is reached, a new (random) target is generated.

Whenever a wick brightess is modified, the new level is set in the ***leds*** object. Note the use of the global *level* array which acts as an overall brightness control value. As we modify the *level* array in the main cog, it is used by the flame cog. I think this is really cool.

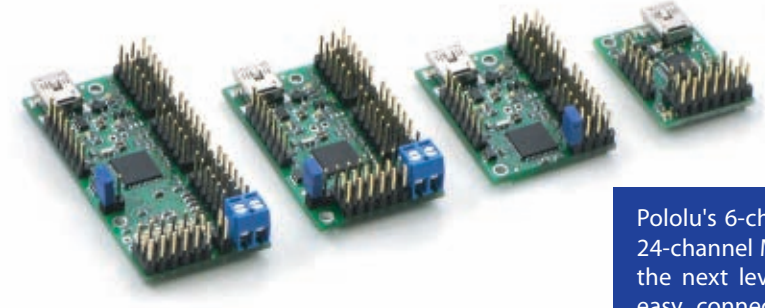
Overall loop speed is dictated by the number of active wicks and the value in *flickerdelay*. As we saw in the main code, we can change *flickerdelay* to alter the apparent behavior of the wicks. You'll see there is just a bit of math before the ***waitcnt*** command — this is needed to prevent ***waitcnt*** from hanging (delay of about 56 seconds) when we have a small value in *flickerdelay*.

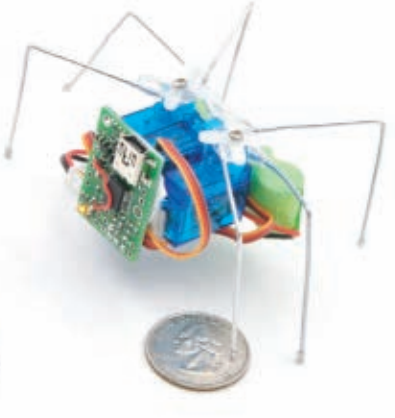
This process could be duplicated in another micro but I believe it would be a much bigger coding challenge.

■ FIGURE 12.
Jack in action.

Introducing Pololu's new line of
Maestro USB Servo Controllers

Conduct a symphony of servos.





- Three control methods: USB, TTL serial, and internal scripting
- Free configuration and control application with motion sequencer
- Channels can be used for digital I/O or up to 12 analog inputs
- Individual servo speed and acceleration control for each channel
- Up to 8 KB of internal scripting memory (~3000 servo positions)
- 0.25 us servo pulse resolution with pulse rate up to 333 Hz

Pololu's 6-channel Micro Maestro and new 12-, 18-, and 24-channel Mini Maestros take serial servo controllers to the next level by incorporating native USB control for easy connection to a PC and programmability via a simple scripting language for self-contained, host controller-free applications. Whether you want the best servo controller available or a versatile, general-purpose I/O control board, these compact devices will deliver.

#1356 Mini Maestro 24-Channel $49.95	#1354 Mini Maestro 18-Channel $39.95	#1352 Mini Maestro 12-Channel $29.95	#1350 Micro Maestro 6-Channel $19.95

Pololu Robotics & Electronics

more information at www.pololu.com/maestro

worldmags

Another great thing about the Propeller is that you can very quickly test new code by downloading straight to its RAM (all programs run from RAM). This lets us make and test changes very easily. We can do this by pressing the F10 key in the IDE. Once everything is the way we like it, we can do a permananet download of the final code with F11.

Well, there you have it; a fun Halloween prop and a neat way to safely simulate flames. Want a hellish pit of burning coals in your Halloween display? No problem! Just use red and orange LEDs and set the flicker algorithm to low-and-slow. With a little imagination and a bit of fine-tuning of the flicker parameters, you can simulate most anything that's "burning." Happy Halloween! **NV**

Jon McPhalen is an actor, writer, and electronics enthusiast based in Los Angeles, CA. He portrayed John F. Kennedy in several episodes of The History Channel's "Unsolved History" series and can be seen with Tom Wilkinson and Frances Fisher in the hit comedy, "The Night of the White Pants."

SHOWCASE

MIAC
PICmicro controller
Dealers Wanted
MIAC is an industrial grade PICmicro controller which retails at $180. Features include direct USB download, 8 analog/digital inputs, 4 relays, 4 motor outputs, integrated CAN bus and a free copy of Flowcode V3. We are looking for commercial partners for MIAC – if you are interested then please call us in the UK on +44 1422 252380.
MATRIX
www.matrixmultimedia.com

QKITS.COM
Power Supply: 12/24VDC Load Voltage
30A PWM DC Motor Speed Control
Control the speed of your electric motors or the brightness of your DC light bulbs.
$23.95
Quantity discounts available
1-888-GO-4-KITS

www.Primecell.com
Battery rebuilding service
Dead Batteries ? Don't toss them. Send them to us - our rebuilds are better than original specifications.
Tools
Hilti Skil Milwaukee Panasonic B&D DeWalt Makita All 2-36 Volts
Electronics
Bar Code Scanners Surveying Printers Laptops Photography
Radios
APELCO UNIDEN G.E. ICOM KENWOOD MOTOROLA MIDLAND MAXON YAESU ALINCO
Visit www.primecell.com for important details
24 Hr Secure recorder tel-fax (814) 623 7000
Quotes email: info@primecell.com
Cunard Assoc. Inc. 9343 US RT 220 Bedford PA 15522

Do you know how many watts (YOUR MONEY) are going down the drain from "THE PHANTOM DRAW?"
The KILL A WATT meter is the best way to help you determine your actual energy draw in ON and OFF home appliances.
Only $29.95
To order call 1 800 783-4624
or online www.nutsvolts.com

worldmags

ALL ELECTRONICS CORPORATION

QUALITY Parts
FAST Shipping
DISCOUNT Pricing

CALL, WRITE, FAX or E-MAIL for a FREE 96 page catalog.
Outside the U.S.A. send $3.00 postage.

MICROMINIATURE 12VDC SPDT RELAY

Fujitsu #FBR21D12. 7.5 x 10 x 10mm high. 12VDC 534 Ohm coil. SPDT contacts rated 1A @ 24Vdc, 0.5A @120VAC. Maximum carry current, 2A. Note: PC leads are not stiff. Not suitable for automatic insertion. **CAT# RLY-616**

80¢ each

10 for 65¢ each
100 for 50¢ each

HEADPHONES WITH MIC

"Hands-free" headset for cordless home or office phones and many cell-phones. In-line volume control and mute switch with pocket clip. 2.5mm sub-miniature phone plug.
CAT# HP-211

$5.85 each

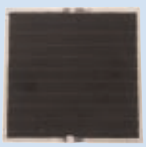
12VDC 750MA POWER SUPPLY

LiteOn#PA-1090-1. 2.5mm coax power plug, center +. Detachable, 2-conductor power cord. cULus, CE.
CAT# PS-12751

40 for $7.75 each

$8.50 each

SOLAR CELL



Output: approximately 3 Volts @ 40 mA. 60mm square x 2.5mm thick epoxy-encapsulated silicon photovoltaic cell. Solid, almost-unbreakable module with solderable foil strips on backside. Ideal for solar-powered battery chargers and other projects.
CAT# SPL-61

100 for $3.25 each

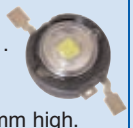
$3.75 each

7-PORT USB HUB

Provides seven additional USB 2.0 ports to your PC or Mac, to connect digital cameras, external hard drives, flash drives, and printers. One upstream and seven downstream ports. Hot-swappable and plug-and-play connections.
CAT# USB-71

$10.95 each

HIGH-POWER COOL-WHITE LED EMITTER

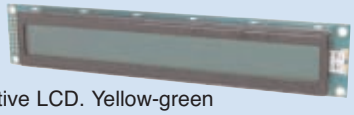


Luxeon Lumiled(TM) LXHL-PW01. Cool-white, Lambertian emitter. 3.42Vdc (typical), 350mA. 30.6 Lumens. 8mm diameter x 6mm high. Surface-mount leads. Spec sheet available.
CAT# LED-145

$2.85 each

10 for $2.75 each
100 for $2.55 each
250 for $2.20 each

40 CHARACTER X 2 LINE LCD W/ LED BACKLIGHT



#UMSH-3077JD-YG. STN Transflective LCD. Yellow-green LED backlight. Module size: 182 x 33.5 x 13.6mm. Viewing area: 154.4 x 16.5mm. Character size: 3.2 x 5.55mm
CAT# LCD-3077

10 for $7.50 each

$8.50 each

NIMH AA CELL, 2500 MAH

High energy, high capacity rechargeable NiMH battery. 1.2V. 2500mAH. 14.5mm diameter x 50mm long.
CAT# NMH-AA

$2.95 each

8 for $2.75 each
60 for $2.40 each

12VDC SOLENOID

Guardian # T4X12-INT-12DC / A420-066012-00. 12Vdc, 24.7 Ohm coil. 6.1W, 486mA. 0.5" dia. x 1.5" long frame. 3/8-32 threaded bushing is 0.3" long. 7" leads. 0.188" dia. slotted plunger with hole. Intermittent duty: 25% on time. 100 seconds max., followed by 300 seconds off min. Includes hex nut, lock washer and mounting bracket.
CAT# SOL-412

$4.00 each

METAL BOX W/ HINGED LID

Good looking metal storage / display box. Looks like brushed aluminum, but responds to magnetism.
CAT# MB-105

45¢ each

10 for 35¢ each

21MM PIEZO ELEMENT

50mm wire leads.
CAT# PE-56

$1.00 each

10 for 85¢ each

12 VDC 58 RPM MINI-MOTOR

12Vdc gearhead motor. 58RPM, @ 12Vdc, 20mA (no-load). 12mm diameter x 35mm long. Solder-lug terminals. 2mm diameter x 6mm long flatted shaft.
CAT# DCM-318

$12.95 each

VIBRATING MINI-MOTOR

Designed to vibrate in cell phones or pagers, this tiny 1.5-4.5Vdc motor has an offset weighted shaft. Overall length, including terminals and shaft, 18mm. Body is 4.5mm x 5.5mm x 11mm long. Flat, surface-mount solder tabs.
CAT# DCM-382

$1.25 each

10 for $1.10 each • 100 for 95¢ each

Shop ON-LINE www.allelectronics.com
ORDER TOLL FREE 1-800-826-5432

MAIL ORDERS TO:
ALL ELECTRONICS CORP.
14928 OXNARD ST., VAN NUYS, CA 91411-2610

FAX (818) 781-2653 • INFO (818) 904-0524
E-MAIL allcorp@allcorp.com

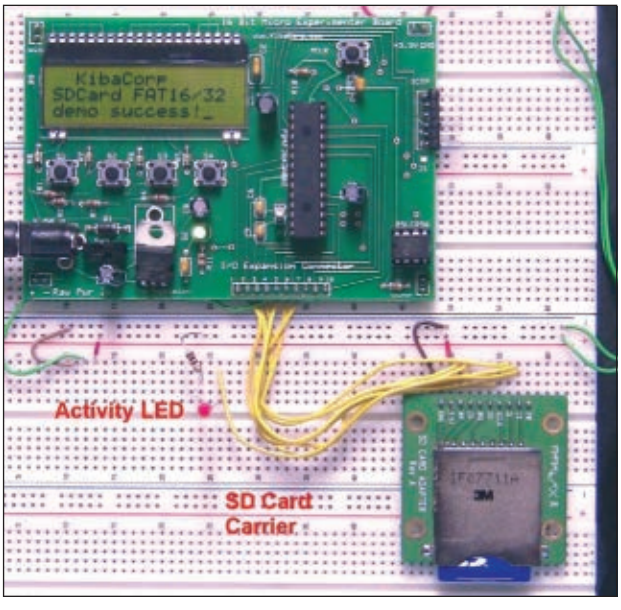
NO MINIMUM ORDER • All Orders Can Be Charged to Visa, Mastercard, American Express or Discover • Checks and Money Orders Accepted by Mail • Orders Delivered in the State of California must include California State Sales Tax • NO C.O.D • Shipping and Handling $7.00 for the 48 Continental United States - ALL OTHERS including Alaska, Hawaii, P.R. and Canada Must Pay Full Shipping • Quantities Limited • Prices Subject to change without notice.

MANUFACTURERS - We Purchase EXCESS INVENTORIES... Call, Write, E-MAIL or Fax YOUR LIST.

BY THOMAS KIBALO

IMPLEMENTING A FILE I/O SYSTEM FOR THE 16-BIT MICRO EXPERIMENTER

In a previous *Nuts & Volts* article, we showed a Graphics Module add-on for the 16-bit Micro Experimenter (Experimenter for short). In this article, we will discuss adding more capability to our Experimenter by integrating a PC compatible file I/O. The SD card technology and Microchip's freely available file I/O library, along with the Parallax SD carrier board makes this all possible. The software and hardware footprint is fairly minimal making the file I/O system an easy add-on module for your Experimenter applications. Gone are the days when you have to labor with custom solutions to move data between your PC and microcontroller. Also gone are the days when you are constrained in the amount of data you can transfer. With this new capability, you essentially have a low cost, small form factor multi-GB removable "hard drive" that can work seamlessly with either a PC or the Experimenter. It will allow both devices to independently create, write to, and read from, delete, and rename data files. This article will discuss the Microchip library briefly, how it was modified for the Experimenter, and the specific hardware and software needed for the Experimenter. It will also cover several demo applications. In this, as well as all other previous articles on the Experimenter, software is written in 'C' so some knowledge of its basic syntax will be required to understand the file I/O library and its API.



■ FIGURE 1. Experimenter with SD card.

Figure 1 shows the Experimenter connected to a 2 GB SD card through the Experimenter I/O expansion port running one of the demo applications (DEMOREADWRITE). It uses an SD carrier board offered by **Parallax.com** for easy insertion and removal of the SD card within the Experimenter's solderless breadboard environment.

The SD Card

Let's review the SD card. It is difficult these days not to notice Secure Digital (SD) cards (see **Figure 2**). They appear just about everywhere. An SD card is a non-volatile Flash memory card format developed for use in portable devices. It is widely used in digital cameras, digital camcorders, and portable laptop computers to name a few. Its size and the format factor make it ideal for adaptability of use with microcontrollers like the Experimenter.

The format has proven to be very popular. Standard SD card capacities are up to and exceed

worldmags

2 GB. These cards ship preformatted with a FAT16 or FAT32 file system. This is the exact file format for all of Windows' machines (we will discuss this more a little later). Because of this, the SD card can be accessed on virtually any PC with an SD reader. Once inserted into a PC reader, Windows allows the SD card to assume the "identity" of another disk drive within the PC. At that point, Windows' operating system can be used seamlessly with the SD card files, and file data can be read as long as Windows software recognizes the file extension associated with the file. For these demos, I restricted myself to Secure Digital SD cards using card sizes of 256 MB, 512 MB, and 2.0 GB. They are readily available, and my suggestion is to use one for these experiments.

Background on FAT16 and FAT32 File Systems

Just to give some background on this subject, in order to store files, the PC needs a filing system that defines the names of the files, as well as a system to track which sector (512 bytes or smallest traceable data "chunk" in a file system) is stored in which file. In addition to reading and writing data, the system should be able to create, delete, and rename files. The File Allocation Table (or FAT) is used as the filing system or "personal card catalog" that supports these functions for both the PC and the SD card.

The job of FAT is to keep track of all files on a given system. FAT16 is the oldest of the Windows file systems and the 16 refers to the fact that this file system allocates 16 bits to store addresses of each sector. It therefore has a limitation of supporting SD cards of up to 2 GB. FAT32, on the other hand, uses 32 bits as a sector address to support up to 2 *TERA bytes* of data.

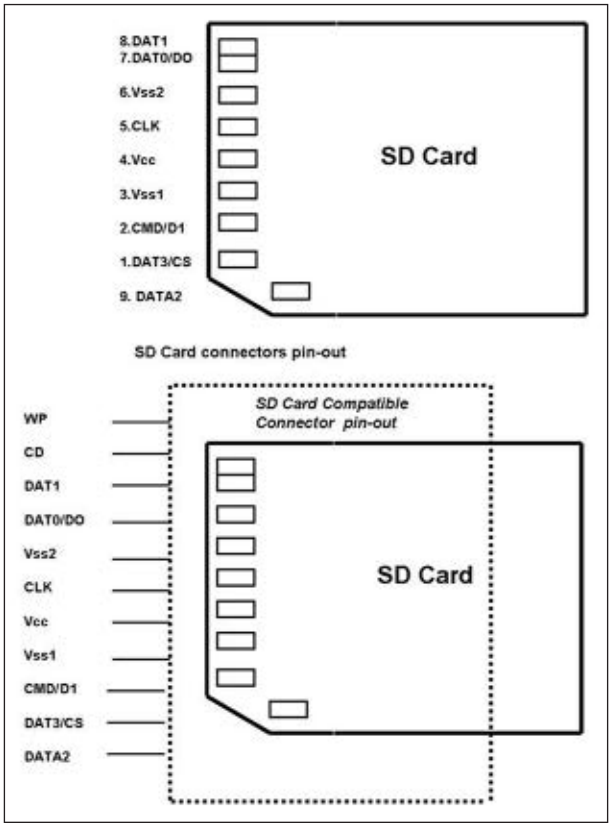
The Microchip library is configured to support both formats. This provides a lot of removable data storage capability for microcontrollers (like the Experimenter) to take advantage of.

The SD Card Hardware Interface

The hardware interfaces are shown in **Figure 3**. The card interface itself is shown first; then the card with the carrier interface second; and finally a block diagram of the entire system connected to the Experimenter. Note that SD cards operate from 2.0V to 3.6V and this is easily accommodated with the Experimenter +3.3V operations. SD cards support nine electrical contacts in addition to two contacts for insertion detection and write-protection switch setting. The nine electrical contacts can work in two distinct modes of communication: the native original SD bus (4-bit parallel interface); or an SPI (Synchronous Serial Interface) mode. SPI is the mode we will use for the Experimenter. When using SPI, not all of the electrical pins are used and are therefore tied high. For the Experimenter, that leaves us with just eight signals in total that we must support (see **Figure 4**).

■ FIGURE 2. SD cards.

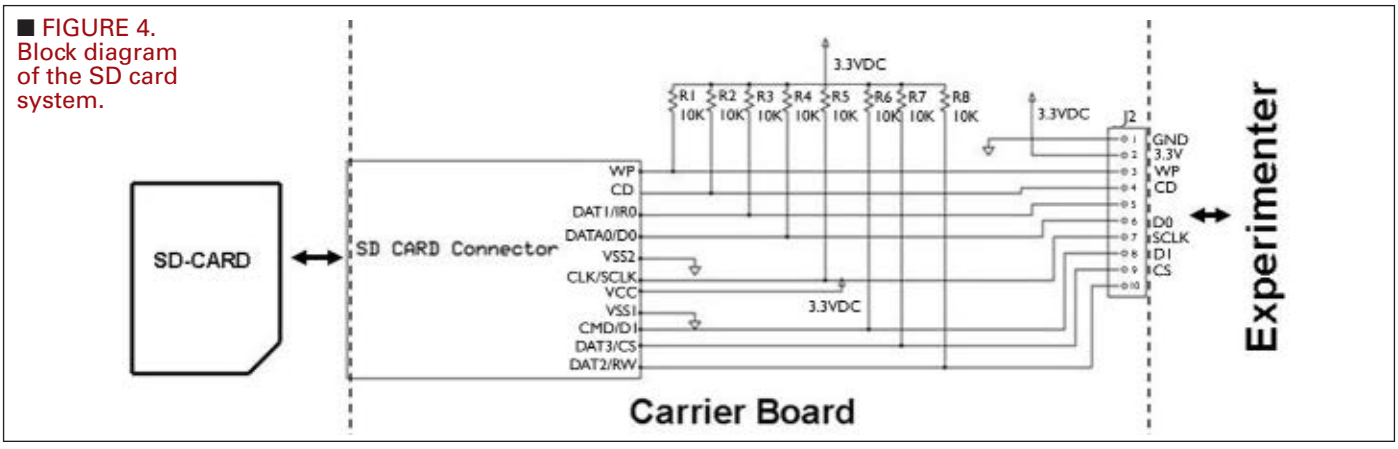
- CD — SD card detect
- CS — Card select
- WP — Card write protection setting
- DO — SPI serial data out of card
- DI — SPI serial data into card
- CLK — SPI clock for data transfer (originates in Experimenter)
- Power, GND — +3.3 VDC and ground from the Experimenter



■ FIGURE 3. SD card and connector interfaces.

The PIC24F microcontroller on the Experimenter has two internal SPI peripherals, SPI1 and SPI2. All the demos configure the Experimenter I/O expansion bus for SPI2 and digital I/O. The SPI2 supplies three basic pins for communication: Serial Data in (SDI), Serial Data out (SDO), and Serial Clock (SCK). Chip Select (CS), card detect, and write-protect are configured as digital I/O. The final I/O expansion bus

worldmags



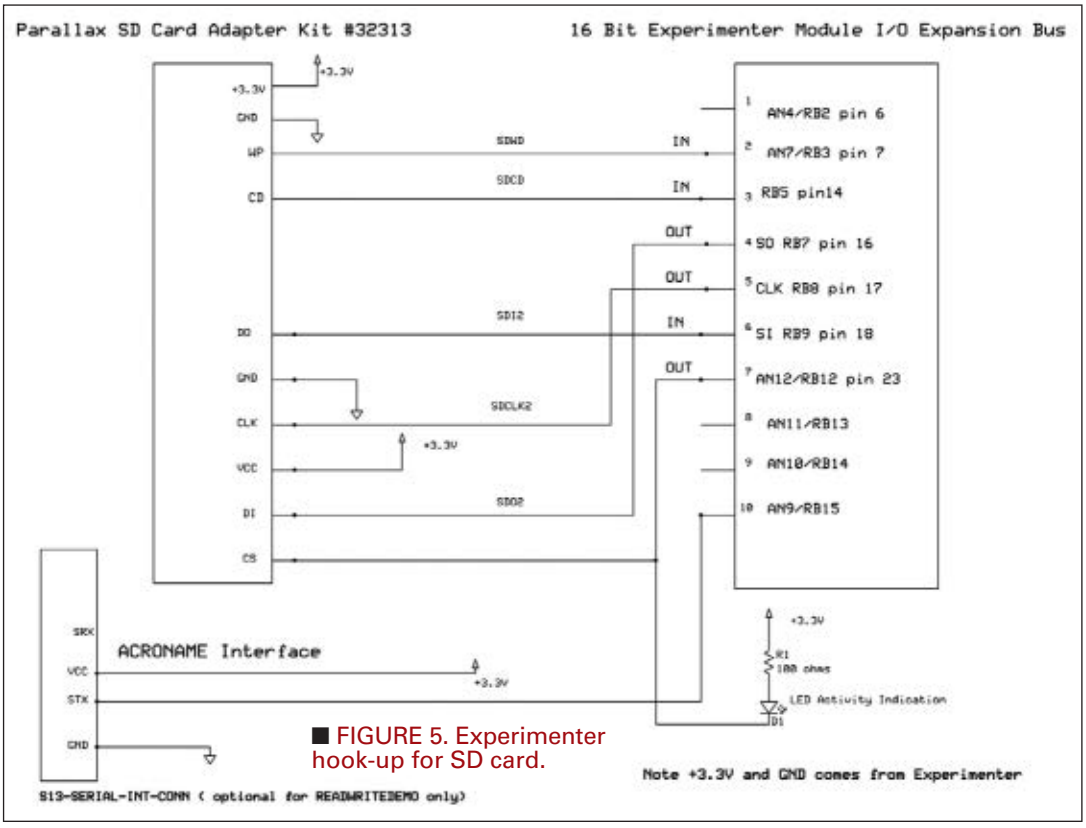
■ FIGURE 4. Block diagram of the SD card system.

interface tally is as follows:

- Pin 4 SPI2 Data Out — Connects to SD card DI.
- Pin 5 SPI2 Clock Out — Connects to SD card CLK.
- Pin 6 SPI2 Data In — Connects to SD card DO.
- PIN 2 Digital In —Connects to (WP) Write Protect (this is low when write-protect is off).
- Pin 3 Digital In — Connects to (CD) Card Detect (this is low when card is present).
- Pin 7 Digital Out — Connects to (CS) Card Select (this is low when card is selected).

A final hook-up diagram using the Parallax SD card adapter and the Experimenter is shown in **Figure 5**. Note that an activity LED is connected to the CS line of the SD card. When the LED is "on," the SD card is actively being accessed from the Experimenter. When the LED is on, don't remove it from the carrier; it is indicating that the Experimenter is actively accessing the card.

To use the SPI2 within the PIC24F, we need to configure the PPS or Programmable Peripheral Select capabilities of the PIC24F. PPS exists as a feature for the smaller pin packages of the Microchip PIC24F family. It allows users to configure where they want the internal peripherals to appear on the outside pins of the package. This is done for the simple reason that the internal peripheral count is too large to be accommodated by the limited pin count of the package. For the PIC24F, there are a total of 16 programmable pins on the package. Ten of these are available for use on the Experimenter's I/O expansion port. The details of this PPS programming can be seen in the main code of each of the demo projects. Details of how PPS is done in general can be found in the datasheet available from Microchip on the PIC24FGA004 series, as well as in the beginner's experiments lesson plans available at **www.KibaCorp.com.**



■ FIGURE 5. Experimenter hook-up for SD card.

Microchip File I/O Library

Microchip developed a file I/O library that is freely available to all users of Microchip products. It is configurable and allows users to only compile and accept those library

worldmags

elements that they want to use in their application. An excellent resource is the Microchip Application Note AN1045 "Implementing File I/O Functions Using Microchip's Memory" that is available on their website at **www.microchip.com**. We adopted this library for the Experimenter, and the applicable API (Application Programming Interface) calls shown below:

- **FSInit** — Initializes card and interface.
- **FSfClose** — Updates file info and closes file.
- **FSfeof** — Verifies if the end of the file has been reached (used in read operation).
- **FSfopen** — Opens existing file or creates new one.
- **FSfread** — Reads opened file.
- **FSfseek** — Changes the file pointer position in file.
- **FSftell** — Returns current pointer position in file.
- **FSfwrite** — Writes information to file.

There are two other functions we need to consider:

- **MDD_MediaDetect** — Determines if SD card is in carrier.
- **SetClockVars** — This API sets the timing variables manually that are used to set the file create/modify/access times. This function is used only if we elect not to use the internal Real Time Clock Calendar (RTCC) peripheral to set time.

Let's go through a couple of examples. There are a total of three demos: simple file write WRITEDEMO; simple file read READDEMO; and a larger example that uses two files (READWRITEDEMO), the RTCC peripheral, UART, and LCD, as well.

```
//modified for 16 bit Micro Experimenter

    // Description: SD-SPI Chip Select Output bit
    #define SD_CS               PORTBbits.RB12
    // Description: SD-SPI Chip Select TRIS bit
    #define SD_CS_TRIS          TRISBbits.TRISB12

    // Description: SD-SPI Card Detect Input bit
    #define SD_CD               PORTBbits.RB5
    // Description: SD-SPI Card Detect TRIS bit
    #define SD_CD_TRIS          TRISBbits.TRISB5

    // Description: SD-SPI Write Protect Check Input bit
    #define SD_WE               PORTBbits.RB3
    // Description: SD-SPI Write Protect Check TRIS bit
    #define SD_WE_TRIS          TRISBbits.TRISB3

    // Registers for the SPI module you want to use

    // Description: The main SPI control register
    #define SPICON1             SPI2CON1
    // Description: The SPI status register
    #define SPISTAT             SPI2STAT
    // Description: The SPI Buffer
    #define SPIBUF              SPI2BUF
    // Description: The receive buffer full bit in the SPI status register
    #define SPISTAT_RBF         SPI2STATbits.SPIRBF
    // Description: The bitwise define for the SPI control register (i.e. _____bits)
    #define SPICON1bits         SPI2CON1bits
    // Description: The bitwise define for the SPI status register (i.e. _____bits)
    #define SPISTATbits         SPI2STATbits
    // Description: The enable bit for the SPI module
    #define SPIENABLE           SPISTATbits.SPIEN
```

Map CS, CD, WE controls to I/O Expansion port

Map use of SPI2 to Library

Hardwareprofile.h

■ FIGURE 6. Library Hardwareprofile.h.

this example can help you understand what is involved in adopting these libraries for your own use. Keep in mind that all our demos are preconfigured and can be used "out of the box" as a template for your applications.

The library options are configured in FSconfig.h by selecting which macros you want to enable. For the Experimenter, the following is enabled:

```
#define FS_MAX_FILES_OPEN  1 :      sets maximum of files open =1
#define MEDIA_SECTOR_SIZE  512 :    sets a  sector size compatible with FAT
#define ALLOW_WRITES:               allows file writing
#define ALLOW_FORMATS :             allows formatting
#define SUPPORT_FAT32  :            sets support for both FAT16 and FAT32 file formats
 #define USERDEFINEDCLOCK:          allows use of SetClockVars () to set manually to time
stamp for file creation and update
#define USEREALTIMECLOCK :          allows use of  PIC24F RTCC (Real Time Clock Calendar)
to automatically set time stamp for file creation and update
```

The hardware interface for the library is configured in two places. The first place is Hardwareprofile.h. The other place is in Main (). Here, we map SPI2 using PPS (described earlier) to the appropriate Experimenter I/O pins. Note that

Configuring the Microchip File I/O Library for the Experimenter

Before we proceed, let's exam how the library was configured to run on the Experimenter. This section let's us look "under the hood" so to speak. Microchip offers lots of free libraries and working through

```
#include "FSIO.h"       //file I/O library
char sendBuffer[] = " Experimenter I/O is a snap!";
char receiveBuffer[50];

int main (void)

{

__builtin_write_OSCCONL(OSCCON & 0xbf) ; //unlock
//input
    RPINR22bits.SDI2R =  9;  // make Pin RP9 SDI2 pin 6 of Expansion I/O

//output
        RPOR3bits.RP7R =10;   //SDO2  RP7 pin 4 of Expansion I/O
        RPOR4bits.RP8R = 11; // SCK2OUT RP8 pin 5 of expansion I/O

__builtin_write_OSCCONL(OSCCON | 0x40) ; //lock

 FSFILE * pointer;
```

file I/O library reference

unlock for PPS

SPI2 PPS programming to pins

lock PPS

In MAIN () Setting up Library for Use with Experimenter

■ FIGURE 7. Library reference and hardware configuration in Main.

worldmags

```
FSFILE * pointer;


while (!MDD_MediaDetect());
     // Step 1 media detected

// Initialize the library
while (!FSInit());
     // Step 2  Microchip file library and SPI initialized

pointer = FSfopen ("FILE1.TXT", "w");
 // Step 3 Create a file to write data to
if (pointer == NULL)
     { // Step 3 error opening first file
       while(1);
     }
     // Step 3 successful opening of first file"

// Step 4 write data to the file
if (FSfwrite (sendBuffer, 1,sizeof sendbuffer, pointer) != sizeof sendbuffer)
     { // Step 4 error writing to first file
    while(1);
     }

     //Step 4 successful write to first file
 //Step 5 closing file
   if (FSfclose (pointer))
   {
     //Step 5 error closing file
    while(1);
   }
    // Step 5 successful file closure
   while (1); // Done writing data to file
 // test remove SD-Card and check contents of your SD-CARD on your PC
```

Step 1
Step 2
Step 3
Step 4
Step 5

■ FIGURE 8. Write demo code.

there is an un-lock and lock sequence that surrounds this PPS programming to insure that this critical function is not invoked haphazardly. Also take note that only #include FSIO.h is needed in Main () to reference the entire Microchip library.

DEMOWRITE - Open a Text File and Write Data to It

Our first example creates a file and writes the following line to it: "**This files I/O test using Microchip library.**" The LCD library is used to display the status. Download and unzip WRITEDEMO from the website. Open the folder and double-click on MDDFS_SD_PIC24.mcp. In Step 1, we call **MDD_MediaDetect ()**. This function determines if the SD card is present. After Step 1, we then execute Step 2: **FSInit ().** This function initializes the library and the SPI port. Step 3 opens a file "FILE1.TXT" with write or the "w" attribute. In this step, the **FSfopen ()** will create the file if it doesn't exist; if it does exist, it will clear it of previous data. Step 4 does the actual write of data to the file using **FSfwrite ()**. This function takes data preloaded in the send buffer array and writes it to the file one byte at a time. In Step 5, we finally close the file with **FSfclose ()**. After executing WRITEDEMO, we can remove the SD card from the Experimenter and place it in the PC SD card reader to verify the file contents. With the SD card in the PC, it will register the card as a hard drive.

■ FIGURE 9. Write demo LCD.

Open up the hard drive and note that FILE1.TXT exists; double-click on the file (Windows Notepad should automatically open it) and the proper text should be displayed. It should read "**This a file I/O test using Microchip library**."

We can create files with any extension that Windows can recognize. However, .TXT is the simplest. This format works well for data written in an ASCII format. If you write lots of data records to your file, keep in mind that you need to use some kind of delimiter between your data records like a space or comma.

■ FIGURE 10. Read demo code.

```
   // set time manually --march 5, 2010 7:21:15 AM
SetClockVars( 2010,3,5,7,21,15);
  while (!MDD_MediaDetect());

  // Initialize the library
  while (!FSInit());
   // Create a file
  pointer = FSfopen ("FILE1.TXT", "r");
  if (pointer == NULL)
   {
      while(1);
   }
  // read into receiveBuffer 1-byte objects from  file
  if (FSfread (receiveBuffer,sizeof sendBuffer,1, pointer) != 1)
   {

      while(1);
   }

  // Close the file
  if (FSfclose (pointer))

 {
      while(1);
   }
   //display read data on LCD
   clear_display();
   position_cursor(0);
   int i;
   for ( i =0; i < sizeof sendBuffer; i++)
write_character_LCD(receiveBuffer[i]);


  while(1);
}
```

Step 1
Step 2
Step 3
Step 4
Step 5
Step 6

■ FIGURE 11. Read demo LCD.

worldmags

DEMOREAD - Open a Text File and Read Data From It

Okay, so we were able to write using the Experimenter and read using the PC, but let's double-check that the Experimenter can read the file it created. Download and unzip READDEMO from the website. Open the folder and double-click on MDDFS_SD_PIC24.mcp. Again, the LCD library is used to display the status. Let's remove the SD card from the PC and put it back in the Experimenter SD carrier board. Note the use of the SetClockVars () in the beginning of the code. Steps 1 and 2 are identical to the earlier write code. In Step 3, we open the file for reading using **FSfopen ()**, using the FILE1.TXT file name, and an "r" or read attribute. In Step 4, we do the actual reading with **FSfread ()**. The entire content from the file is read one byte at a time and loaded into a receiveBuffer. We then close the file in Step 5. In Step 6, the LCD display is used to show the entire contents in receiveBuffer**.** You can exam the LCD for a complete string readout for "**This a file I/O test using Microchip library**."

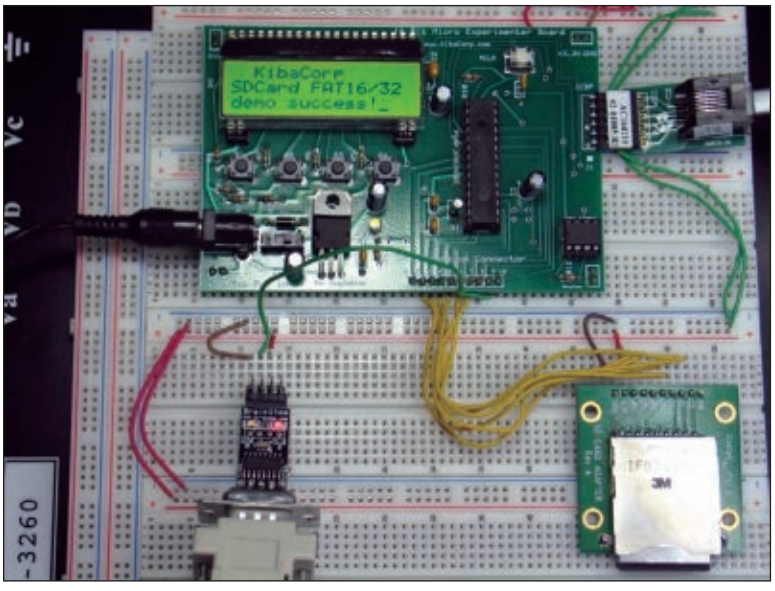
DEMOREADWRITE - A Larger Example Using Two Files, File Pointers, RTCC, LCD, and Serial I/O

This demo creates two files (one at a time) and uses the RTCC to set the date/time of file usage. In addition, both the UART and LCD libraries are used. The UART is totally optional but does require an Acroname adapter (covered in *NV* June '10 issue) and setting HyperTerminal to 9600 8N1 (see the hook-up diagram shown earlier) to display the contents on your PC. Download and unzip READWRITEDEMO from the website. Open the folder and double-click on MDDFS_SD_PIC24.mcp.

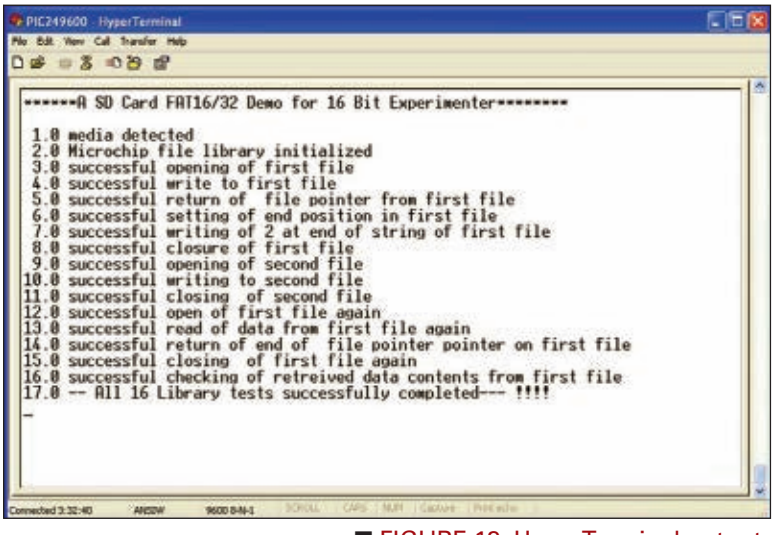
This demo first writes a "This is Experimenter String1" to FILE1.TXT; it then copies this to FILE2.TXT and rewrites the last character in FILE1.TXT to a 2. The results are shown in **Figure 14**. All steps are displayed on the Experimenter LCD simultaneously as they're being written to HyperTerminal.

Future Expansion

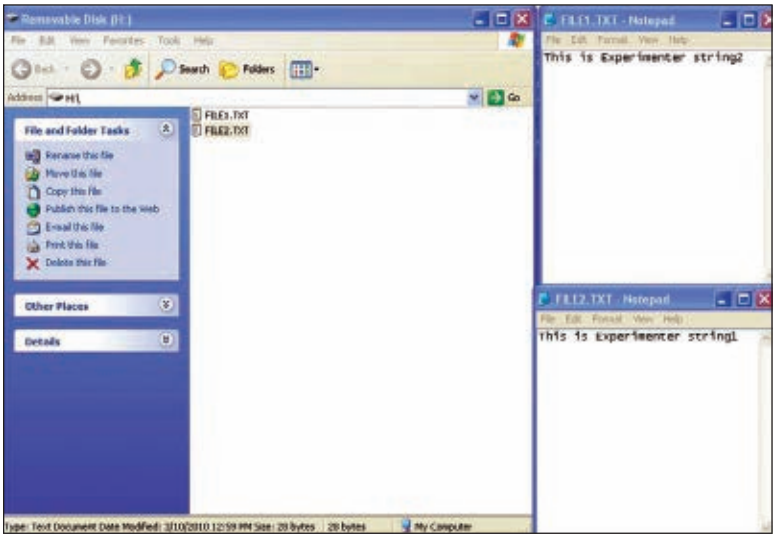
We've now added a pretty substantial capability to our Experimenter. Data logging and data sharing have reached new heights — we now have GB storage and retrieval capability in a microcontroller environment and the ability to use Windows-based tools seamlessly with the data. Think of the applications! **NV**



■ FIGURE 12. Read and write demo with serial port interface shown.


PIC24F600 - HyperTerminal
File Edit View Call Transfer Help
-------A SD Card FAT16/32 Demo for 16 Bit Experimenter---------
1.0 media detected
2.0 Microchip file library initialized
3.0 successful opening of first file
4.0 successful write to first file
5.0 successful return of file pointer from first file
6.0 successful setting of end position in first file
7.0 successful writing of 2 at end of string of first file
8.0 successful closure of first file
9.0 successful opening of second file
10.0 successful writing to second file
11.0 successful closing of second file
12.0 successful open of first file again
13.0 successful read of data from first file again
14.0 successful return of end of file pointer pointer on first file
15.0 successful closing of first file again
16.0 successful checking of retreived data contents from first file
17.0 -- All 16 Library tests successfully completed--- !!!!

■ FIGURE 13. HyperTerminal output.



■ FIGURE 14. Disk contents.

worldmags

PHREAK OUT WITH THE BASIC STAMP 2

By L. Abraham Smith N3BAH

For many years, hackers built special electronic devices known as "blue boxes" for the purpose of transmitting special audio tones over telephone lines. The practice — known as blue boxing — provided challenges to the subcategory of hackers known as phone "phreaks." They had to be able to both construct the tone oscillator circuits to produce the telephone company's proprietary audio tones, and devise how to employ them to explore the telephone switching networks. While the practice of blue boxing died out in the 1990's with the introduction of electronic switching systems which were not susceptible to in-band tone manipulation, interest in the technical aspects of telephone communications continued.

Now, a new generation of telephone experimenters are phreaking out over Asterisk™ (**www.asterisk.org**) — the open source Linux program which can put your own private telephone system inside your computer! The proliferation of computers running Asterisk interconnected over the Internet and interfaced to the public switched telephone network (PSTN) has given rise to a new form of legal phreaking where users configure their Asterisk servers with special features which can only be accessed by users transmitting the old-school blue box tones in appropriate combinations. This project which has gained worldwide popularity is known as Project MF (**www.projectmf.com**). Briefly stated, Project MF recreates the techno environment which existed "back in the day," making it possible to experience the challenges of blue boxing from the legal safety of the Asterisk VOIP system.

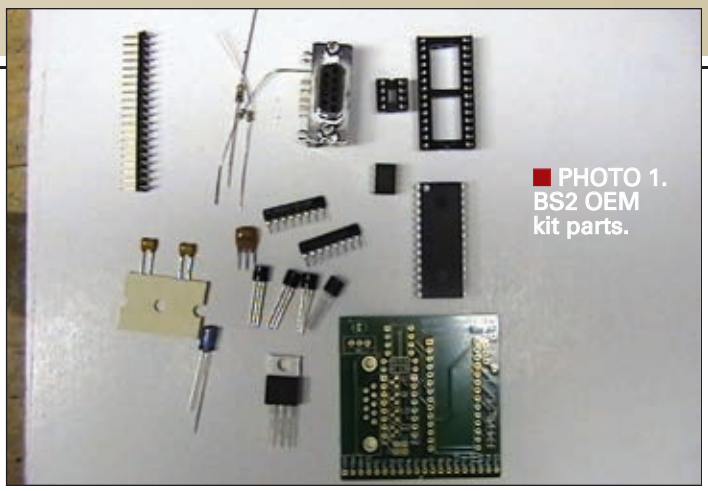
Recently, I began experimenting with Asterisk and Project MF. I needed a low cost programmable audio tone generator capable of producing standard Dual-Tone Multi-Frequency (DTMF) telephone touchtones, as well as unique Single- and Multi-Frequency (MF) tones. I wanted to design a circuit using readily available components which was easy to assemble and which could be modified by firmware changes as my needs evolved. The solution I implemented uses a microcontroller and provides the novice builder with an inexpensive and fun way to learn about the interaction of micro hardware and software while constructing a unique and useful tone generation device.

Why a Microcontroller?

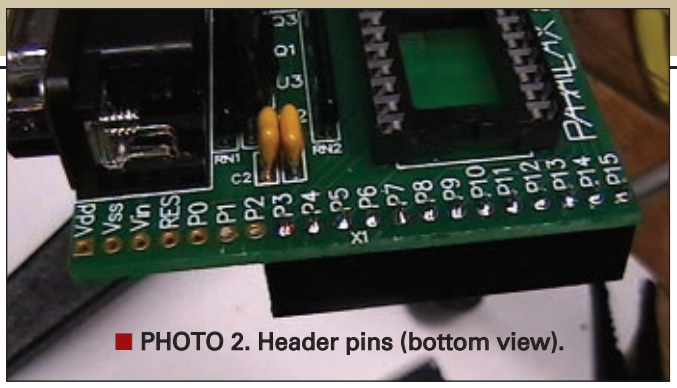
While there is no shortage of PC software which could generate the necessary tones, what fun is that? In order to capture the full effect of the retro-tech environment presented by Project MF, you need to build a hardware blue box. I found a number of schematics on the Internet, most of which either required obscure tone generator ICs or set the tones with tedious potentiometer-tuned circuits. I wanted a circuit which would use readily available parts and allow tones to be easily programmed and changed via embedded software. The desire for a combination of hardware and modifiable software in one package led me to consider a microcontroller.

Most microcontrollers share basic features with each manufacturer's products having their own unique instructions or programming objects which make them better suited for certain tasks. My experience has demonstrated that the BASIC Stamp 2 from Parallax, Inc., is uniquely suited for the purpose of audio frequency generation. There are less expensive micro platforms than

worldmags



■ PHOTO 1. BS2 OEM kit parts.



■ PHOTO 2. Header pins (bottom view).

the BS2, but I chose this module because it is strongly supported with free tutorials and programming documentation from the Parallax website and numerous online user groups. It programs in PBASIC which is easily learned by hobbyists who are not experienced in programming. After you're finished experimenting with the project described here, you will have the confidence and skills to re-use the BS2 in other projects with programs you create!

With the BASIC Stamp, FREQOUT and DTMFOUT do all the work

The kit for the BASIC Stamp 2 OEM module (Parallax part number 27291) is easy to assemble even for the novice builder, and includes a PCB (printed circuit board) with all the components and their locations clearly marked. It is easily programmed via the on-board serial port in the PBASIC language developed by Parallax. One of the unique features of the BS2 is its ability to generate standard touchtones, as well as custom MF tones using the PBASIC commands DTMFOUT and FREQOUT which make it possible to generate both standard telephone Touchtones as well as any other single- or dual-frequency tones and sequences. This micro module is well supported with free reference materials and IDE (Integrated Development Environment), and serves as a good intro to microcontrollers and their practical uses (which will doubtless spur the user to devise project ideas of their own). Finally, the BASIC Stamp 2 is available from a number of sources.

Here you'll learn how to interface the BS2 module with a keypad, a few switches, and a handful of discrete components, then load free open source firmware to make your own unlimited tone generator box. The concept is simple: The hardware checks the state of the 12 keys on the keypad in a continuous cycle, along with monitoring the state of the toggle and pushbutton switch; it then instructs the BS2 to generate the appropriate touchtone or custom tone(s) output via pin 0. By simply changing the values in the software and loading it onto the BS2 via the integral serial port, you can generate any single- and multi-frequency tone(s)/sequences without changing the hardware. Tone frequency, duration, pauses, and sequences can all be modified according to your needs. (More on the programming later.) Now, let's build the PhreakEncoder!

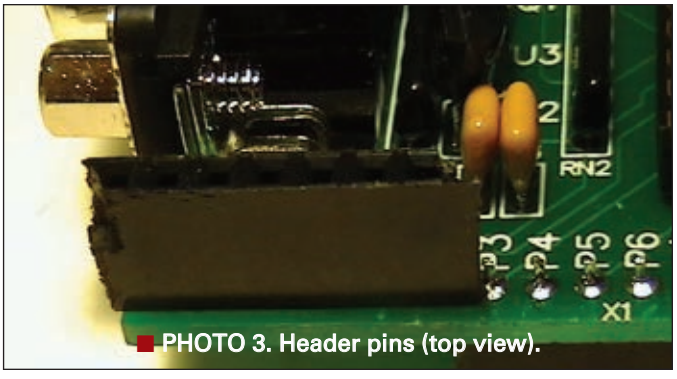
The Build

Step 1: Assemble the BS2.

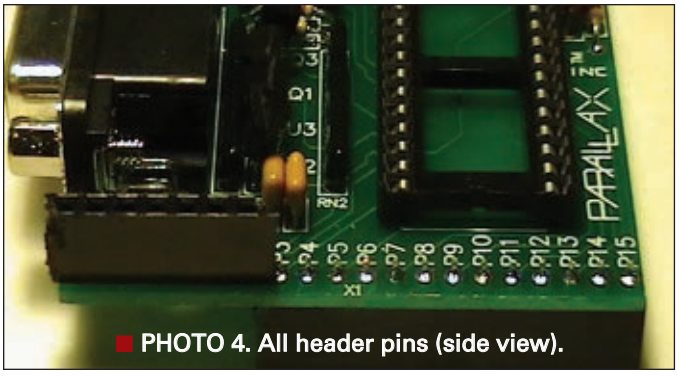
The BS2 OEM version comes complete with the PCB and all necessary parts, and costs about $30 for the kit form. (See **Photo 1**.) Assemble the module according to the directions from Parallax (**www.parallax.com**); however, do not install the bank of 16 angled male header pins on the lower edge of the board. This will permit direct wired connections between these BS2 pin connections and the keypad and control switches later.

There are several approaches you can use to assemble this project. You can breadboard the BS2 and related components if you intend on building the project as a temporary setup. Or, if you'd like a more permanent assembly you may want to house everything in a project case like I did.

I chose to sandwich the BS2 module and keypad on opposite sides of the project case lid. I soldered in a divided row of 16 female header pins on the output tabs of the BS2

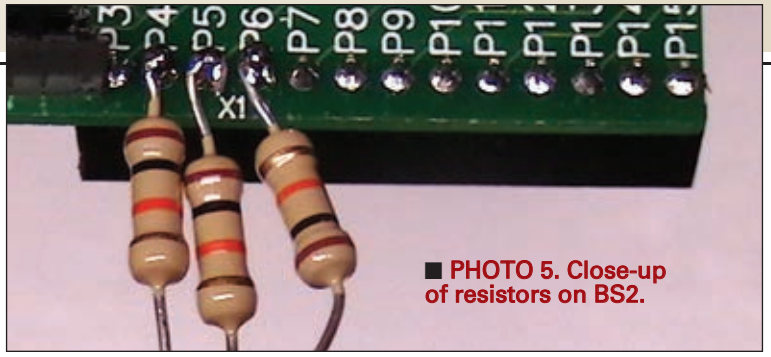


■ PHOTO 3. Header pins (top view).

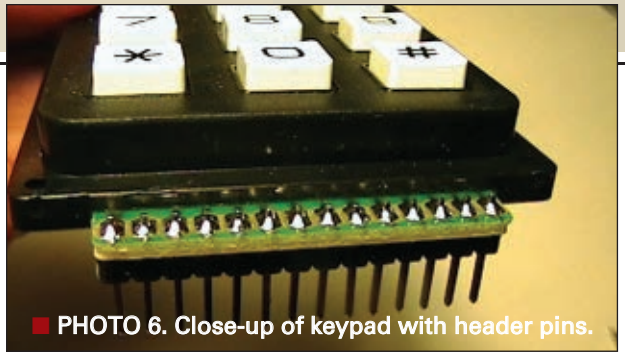


■ PHOTO 4. All header pins (side view).

worldmags



■ PHOTO 5. Close-up of resistors on BS2.



■ PHOTO 6. Close-up of keypad with header pins.

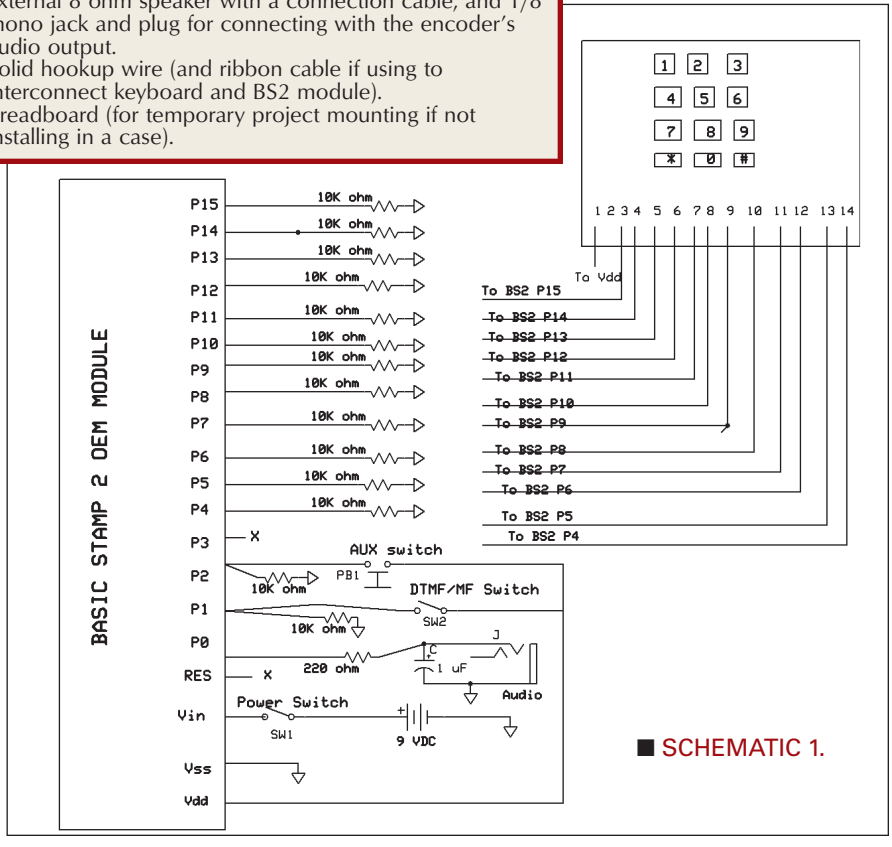
module. For outputs marked Vdd, Vss, Vin, RES, P01, P02, and P03, install the female header pins on the top (component) side of the PCB. All remaining outputs (P04 through P15) had the female header pins installed on the underside of the PCB (See **Photos 2, 3** and **4**.)

At this point it's a good idea to test out your BS2 module. First download and install the most recent version of the BASIC Stamp Editor (BSE) from **www.parallax.com/tabid/441/Default.aspx**. You will need this in order to transfer programming code from your computer to the BS2 during construction, as well as to download your own modified code later on. Connect the positive terminal of the nine-volt battery to the Vin pin, and the negative terminal to the Vss pin. Connect the BS2 to your PC via either a DB9 serial cable or a supported serial-to-USB adapter. Once the BSE recognizes your module, you can test it out with a simple LED flasher circuit.

Connect the anode of an LED to pin 5 and connect the cathode to the Vss (ground) pin via a 220-ohm resistor. Then, download and install the LEDTest.bs2 program to the BS2 module (programs are available for download at **www.nutsvolts.com**). The LED should flash approximately once per second. If it doesn't, check all connections on the module, as well as the LED test circuit, and fix any wiring errors. Once the module tests okay, remove the LED, 220-ohm resistor, and power connections, and disconnect the PC interface cable.

PARTS LIST

- BASIC Stamp 2 OEM module kit ($30.95; **Parallax.com**, part # 27291).
- Velleman 12-key keypad (**AllElectronics.com** part # KP-12, or from **electronicpartsforless.com** or **willyselectronics.com** part # 12KEY.) NOTE: The keypad must have separate connectors for each of the 12 keys, plus a common connector; a 3x4 matrix type keypad is NOT acceptable.
- (14) 10K ohm resistors, 1/4 watt.
- 220 ohm resistor, 1/4 watt.
- 1 µF electrolytic capacitor.
- (2) SPST toggle switches.
- Female breakaway headers, 16-pin strip (**Sparkfun.com**, SKU# PRT-00115); (Optional – only needed if using a project case and mounting keypad on outside.)
- SPST pushbutton switch, N.O. contacts.
- 9V battery clip with leads.
- External 8 ohm speaker with a connection cable, and 1/8 mono jack and plug for connecting with the encoder's audio output.
- Solid hookup wire (and ribbon cable if using to interconnect keyboard and BS2 module).
- Breadboard (for temporary project mounting if not installing in a case).



■ SCHEMATIC 1.

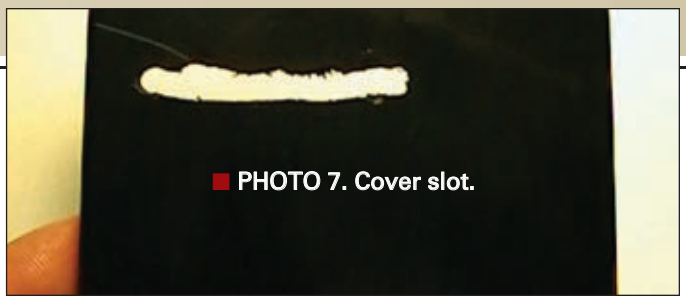
Step 2: Add resistors.

Once you have the BS2 module assembled and tested, solder a 10K ohm resistor to each of the pins labeled P1, P2, and P4 through P15 on the underside of the board. As shown in **Photo 5**, be sure to keep the lead from the board side of each resistor as short as possible. Solder the loose ends of these 14 resistors together via a short length of wire, forming a common connection to the BS2 module's grounding pin (Vss). This is necessary in order for the microcontroller to function properly, as any switches connected to its I/O pins must be pulled low via resistors so that they are not floating when in the off (switch open) state.

Step 3: Connect the keypad and switches.

Now, connect the 12-key keypad to the appropriate BS2 module pin connections (see **schematic**.) This may be accomplished via ribbon cable or other point-to-point wiring. Use adequate lengths of wire between the

worldmags

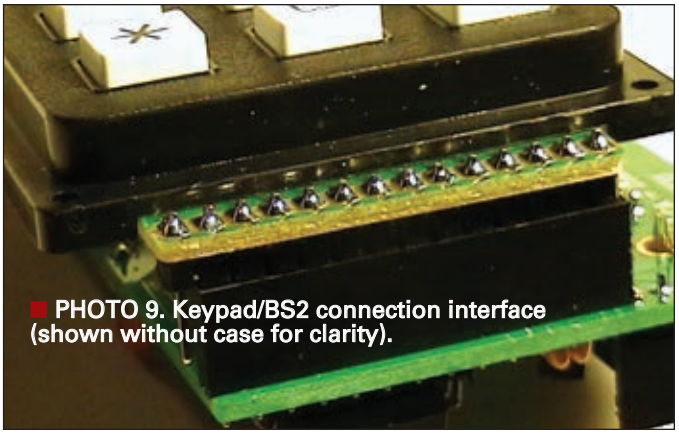


■ PHOTO 7. Cover slot.

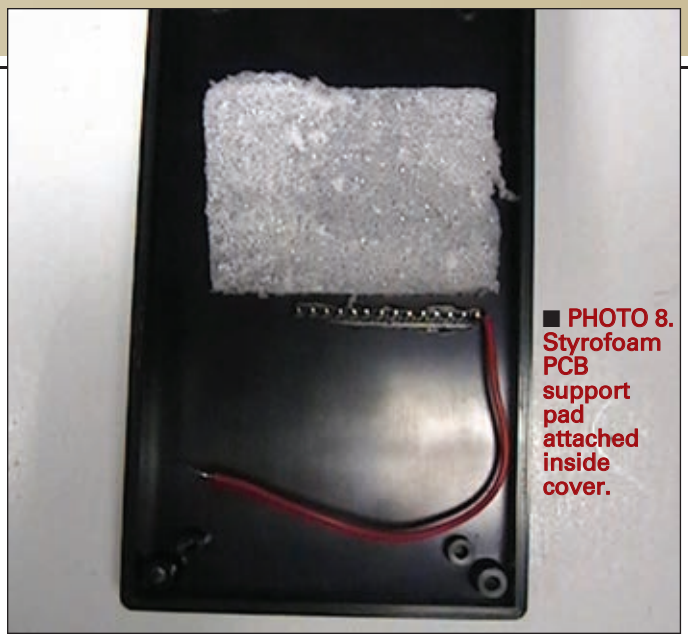
module and keypad, based upon how you will be mounting the finished project. Note that you must use a keypad with individual connections for each of the 12 keys and one common connection; I recommend using the Velleman 12-key keypad. Since I intended to house everything in a project case, instead of using ribbon cable I soldered a set of 14 male header pins to the underside of the keypad (i.e., the side without the keys). **Photo 6** shows these header pins will connect to the BS2 module pins 4 through 15 on final assembly.

Next, solder an adequate length of hookup wire to pin 1 of the keypad to allow it to connect to Vdd on the BS2. The power switch is an SPST toggle connected between the nine-volt battery's positive terminal and Vin. Connect the battery's negative terminal to Vss. The DTMF/MF switch is an SPST toggle connected between P1 and Vdd. The Aux button is an SPST N.O. pushbutton switch connected between P2 and Vdd. The Audio output jack's center conductor is connected to P0 via a 220-ohm resistor and the shield ring to ground. Connect the 1 µF electrolytic capacitor's positive leg to the center connector of the Audio jack and the negative side to the Audio ground terminal.

Since I wanted to package the encoder inside a small (5″ L x 3″ W x 2″ H) project case, I connected the keypad and BS2 module by sandwiching them on opposing sides of the case lid and joining them via their respective header pins through a slot I dremeled into the lid. (I attached the back of the keypad to the outside of the lid with a plumber's adhesive known as "Goop;" the BS2 module was supported inside the lid with a thin piece of Styrofoam and glued in place. Be sure that pin 3 of the keypad is connected to pin 4 of the BS2 module, continuing to connect the pins in a row so that pin 14 of the keypad connects with pin 15 of the BS2, leaving pins 1 and 2 of the keypad unconnected (see **Photos 7** through **9**). However you package the hardware, make sure the DB9 connector



■ PHOTO 9. Keypad/BS2 connection interface (shown without case for clarity).

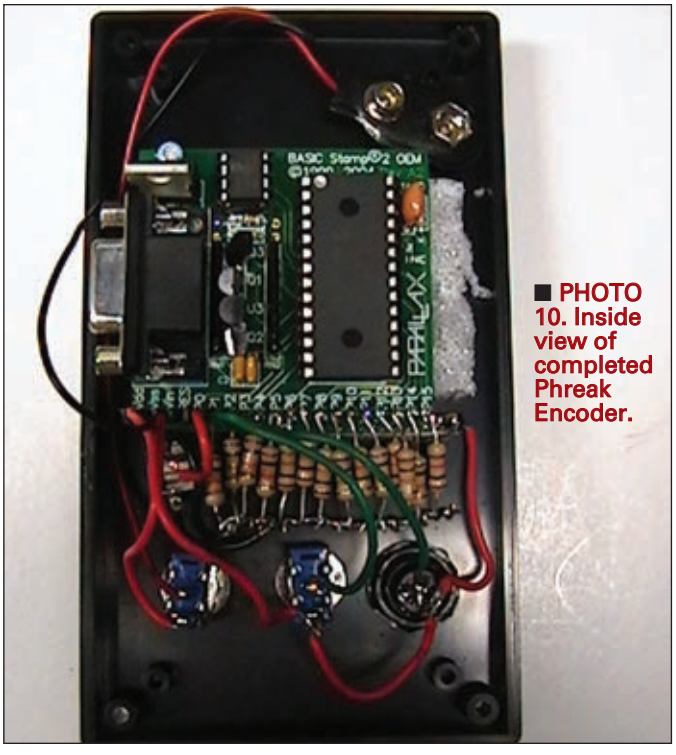


■ PHOTO 8. Styrofoam PCB support pad attached inside cover.

on the BS2 module will be accessible for programming. Assembly of the PhreakEncoder hardware is now complete (**Photo 10**)! Now, finish the build by loading the software.

Load the Code and Test It Out

Connect the PhreakEncoder to your PC via the serial cable and use the BSE to load the PhreakEncoder.bs2 firmware (also on the *NV* website) onto the device. Connect a small eight-ohm speaker of your choice via the Audio output jack and turn on the power. With the DTMF/MF switch in the DTMF position, the keypad will produce standard telephone touchtones of 200 milliseconds duration. (If you do not hear tones in the speaker, check all wiring and connections against the diagram, and be sure



■ PHOTO 10. Inside view of completed Phreak Encoder.

worldmags

■ PHOTO 11. The PhreakEncoder in use.

that the keypad and BS2 pins are correctly joined.) In the MF position, the PhreakEncoder will produce traditional MF/blue box tones formerly used on old telephone switching systems on keys 0-9 (which are needed by Asterisk's ProjectMF see **Table 1**).

When the Aux button is pressed, the keys produce alternate select tones per the chart in **Table 2**. Of course, these tones are for illustration purposes only. The magic of the PhreakEncoder is that you can easily change the open source BASIC Stamp program to produce any single- or multi-frequency tones you want! Sequences of touchtones can be programmed, turning the device into a quick dialer for phone, bank account, or other lengthy numbers you need to send via telephone. You now have an inexpensive, customizable tone machine which can generate unique tones for circuit testing or over-the-air radio control such as in amateur radio repeater control or other remote signaling purposes.

Preprogrammed keypad frequencies contained in PhreakEncoder.bs2.

MF SWITCH POSITION:

Table 1

KEYPAD	DUAL-TONES (IN Hz)
1	700/900
2	700/1100
3	900/1100
4	700/1300
5	900/1300
6	1100/1300
7	700/1500
8	900/1500
9	1100/1500
0	1300/1500

Relationship between keypad and BS2 pins:

Table 3

KEYPAD	BS2
1	12
2	8
3	4
4	13
5	9
6	5
7	14
8	10
9	6
*	15
#	7

Preprogrammed Auxiliary keypad frequencies contained in PhreakEncoder.bs2.

AUX BUTTON PRESSED:

Table 2

KEYPAD	TONE(S) (IN Hz) AND COMMON NAME:
1	2400
2	1300/1700 "KP2 – Keypulse 2"
3	2600
4	1300/1500 "KP - Keypulse"
5	1500/1700 "ST – Start"
6	2200/1700 (pulsed five times alternating with 33 ms of silence) "25 cent tone"
7	697/1633 "A – extra touchtone key"
8	770/1633 "B – extra touchtone key"
9	852/1633 "C – extra touchtone key"
0	941/1633 "D – extra touchtone key"
*	700/1700 "Code 11 – Start 3"
#	900/1700 "Code 12 – Start 2"

Start Phreaking!

The best way to use the PhreakEncoder is to hold the external speaker up to the telephone microphone, or near the microphone on a speakerphone (see **Photo 11**). Then, you would dial into one of the Asterisk Project MF numbers found on the Internet. Usually, the server will provide audio directions as to how to initialize its services.

Hack the Code!

The PhreakEncoder firmware is written using the PBASIC language. When examining the code using the BS2 IDE, pay close attention to those lines containing the DTMFOUT and FREQOUT commands. In order to alter the tones or produce tone sequences, you'll need to modify the arguments to these commands. The syntax for the DTMFOUT command is as follows: DTMFOUT pin_number, [digit1, optional_digit2, optional_digit3] – where pin_number is the desired output pin on the BS2, and digit(s) are the standard touchtone keys 0-9, *, and #.

The syntax for the FREQOUT command is as follows: FREQOUT pin_number, duration, tone1, [optional_tone2] – where pin_number is the desired output pin on the BS2; the duration is the tone duration in milliseconds; and tone1 and optional_tone2 are the desired audio frequencies in Hertz, with values between 0 and 32,767 Hz.

By reading the excellent Parallax support and reference materials available free on their website, you can pick apart the firmware and mod the source code as you desire. Take a look at **Table 3** for a list correlating the BS2 output pins to keypad buttons. You'll need this in modding the code, to cross reference the keypad keys to the BS2 pins needed by the program. Don't forget that pin 3 on the microcontroller is unused in this project. (After reading the reference materials, who knows what mods you'll devise!)

Why Build this Circuit?

You may wonder why you should go through the trouble of building a dedicated circuit just to produce these tones for Project MF. If you remember the old school hacking/phreaking days, then the ability to recreate these activities within a legally safe environment is probably motivation enough. For those of you who weren't active in electronics at that time, you will be able to enjoy the same technical challenges and excitement experienced by hackers of a previous generation. And while not *everyone* who starts out building a blue box ends up founding a major computer company, you just never know. **NV**

worldmags

BY LOUIS E. FRENZEL W5LEF

Q & A ABOUT DIGITAL RADIO

I was talking to a friend of mine recently and he asked if I had ever built a crystal radio. I said yes, and asked why he wanted to know. He answered that he had never built one and just wanted to see if he could. I told him not to delay because pretty soon there would be no radio stations you could receive on a crystal radio.

Crystal radios are just a simple diode demodulator for amplitude modulation (AM) signals. Typically, crystal radios are built to receive local AM broadcast stations. And while those stations still exist, that might not be the case in the future. Why? Because almost all radios today have been converted to digital. AM broadcast stations are one of the few remaining radio services that still use analog AM. What ensued with my friend was a Q & A discussion of what digital radio is and how broad it is. Here is a summary of that discussion.

Q: What do you mean by the term radio?

A: A radio can be a receiver, a transmitter, or (more commonly today) a transceiver. Radio includes AM and FM broadcast, TV, any two-way radios, satellites, cell phones, hobby radios, marine and aircraft radios, military radios, radars, GPS, and so on. Anything we generally refer to as wireless means radio.

Q: How long has this digitization of radio been going on?

A: For decades actually. It probably started with satellites in the 1970's, then moved on to other services over the years. Things really picked up in the 1980's and I would guess that 90% of all radio is digital today.

Q: Just why did this move to digital occur?

A: Great question. And there is no one single reason. Probably the main reason is that digital signals are very robust and less affected by noise because of their binary nature. Second, we now have small, cheap, and fast integrated circuits and microcomputers that can process these signals. Designers found they could squeeze more signals into a given spectrum band with digital techniques thereby greatly improving spectral efficiency in a limited spectrum. Finally, some really great digital compression techniques were developed that further improve the ability to cram more signals at higher speeds into a limited bandwidth.

Q: Give me a good example of how some of those benefits were applied.

A: The best example is cellular radio. Cell phones started out in the 1980's as analog FM signals. Even with a relatively narrow band (30 kHz) modulation, the cell phone carriers found that their limited spectrum was quickly used up meaning they could not take on any more new customers. With profit limited, they because extremely aggressive in creating new digital technologies to fix that problem. Digital radios solved the problem initially by using Time Division Multiplexing (TDM) to put three

worldmags

or (in some systems like GSM which is still in use), eight signals per spectrum channel. This really multiplied their subscriber capacity. Also created around this time was Code Division Multiple Access (CDMA) which further expanded the capacity of the cellular network. Today, all cell phones are digital and a wide range of different technologies are involved.

Q: Are two-way mobile radios also digital?

A: Some of them are but not all. Many like those used in fire, police, and public services still use FM. Digital technologies like P25 and Tetra have been around a while but the different services have been slow to convert. Military radios are mostly digital today.

Q: What about marine and aircraft radios?

A: Strangely, both are still analog. Marine radios use FM and aircraft radios use AM. I'm not sure of all the reasons for that, but I assume that since the ranges are short and the spectrum is not so crowded, why change? Besides, the technologies are well proven and the equipment is moderately priced and very reliable.

Q: What about broadcast radio?

A: Analog AM and FM stations are still around and still the most widely used. However, most stations have added a digital capability called HD radio. The station's programs are digitized and broadcast in digital format on exactly the same frequency as the analog signals. The digital signals overlay the analog signals. Analog radios ignore the digital while an HD radio receiver can pick up the digital signals. The digital capability also allows most radio stations to broadcast two or three other programs simultaneously thanks to the multiplexing capability.

Q: Tell me more about digital HD radio.

A: Well, it has been around for years but in the past five years or so most stations have added it. I am not so sure that most folks actually know about it. You can readily buy an HD radio for about $100 at Best Buy, RadioShack, or any other consumer electronics retailer. Many cars come with HD or offer it as an option. The big deal about HD is that is gives slightly better fidelity than the AM and FM stations. AM stations sound more like FM and FM stations sound more like CDs. Furthermore, the HD signals are more immune to noise and do not fade as much when you are driving.

Q: Does the rest of the world have digital radio?

A: Yes, pretty much. Europe has had digital radio for years with a system called Eureka-147. You will hear it referred to as Digital Audio Broadcast (DAB). It broadcasts music and other programs in digital on the VHF bands (not the usual AM low frequency bands) allotted for that. Canada has a similar system.

Q: What about TV?

A: As you probably know, TV went all digital last year. All the analog stations went off the air and switched to digital which gave us high definition TV. If you still have an old analog set, you have to use one of those converter boxes that translates the digital TV signals to analog so your old set can handle them. All new sets sold today are digital only. Almost everyone loves digital TV as it has such improved resolution. Over 50% of the US population today has an HD set with a big LCD or plasma screen. There is also the mobile TV from FLO TV that you may have heard about. It was developed to provide digital TV to cell phones and portable/mobile TVs like those in cars and SUVs.

Q: Is digital TV unique to the US?

A: No, not really. Europe went to digital TV a number of years ago with its Digital Video Broadcast (DVB) standard. Japan has their ISDB (Integrated Services Digital Broadcast) system and South Korea uses their DMB (Digital Media Broadcast) system. Most of these digital TV standards have mobile versions for broadcasts to cell phones. The US is working on a mobile version of its Advanced Television Standards Committee (ATSC) standard for digital TV in the US. It should be available in the near future in some cell phones.

Q: Are there any radio services that have not moved to digital?

A: Yes, namely the hobby radio services. CB radio is still AM and SSB. The family radio service (FRS) is still FM. So is most ham radio which relies heavily on SSB and FM. Morse coded CW is a form of digital and that modulation is Amplitude Shift Keying (ASK) — a form of AM. Hams also use a form of digital called PSK31. These services lack the spectrum and the FCC restricts their modulation methods, but the movement is in the digital direction.

Q: What about shortwave radio?

A: Shortwave (SW) radio is still mostly analog. Virtually all worldwide stations broadcast in AM. It uses minimum bandwidth in a crowded spectrum but it is susceptible to noise as you know if you listen to SW.

Q: Will shortwave ever go digital?

A: Yes, and it is already happening. A digital standard called Digital Radio Mondiale (DRM) was developed and uses voice compression along with CODFM modulation. It is used primarily in Europe. The result is good quality audio that fits within the assigned narrow (20 kHz or less) SW channels. It's not wide spread yet, but it seems to be catching on. A number of radios are available in Europe but there are few on sale in the US. Some of the new digital software-defined radios have DRM demodulators.

■ FIGURE 1. C.Crane's CC Wi-Fi digital Internet radio connects wirelessly by Wi-Fi to your home router or by a direct Ethernet cable connection.

Q: Are there any other digital radios you haven't told me about?

A: Yes, there are almost too many to mention. For example, most short-range wireless data applications are digital. Some examples are garage door openers, remote keyless entry devices on vehicles, and remote temperature sensors. GPS receivers for navigation are also digital.

Q: What else?

A: Well, the Wi-Fi wireless LAN radios in your laptop or cell phone are digital, of course. These are the radios that let you access the Internet and your email at local hot spots. Bluetooth radios in your car or cell phone headset are all digital. Some wireless speakers for home stereo systems are all digital.

Q: You mentioned the HD radio in cars. What about satellite car radios?

A: Yes, satellite radios in cars are all digital. Sirius Radio and XM Radio merged recently under Sirius but they maintain separate digital satellite systems. Like other digital radios, the sound quality is CD level.

Q: What is an Internet radio? I've heard of that recently.

A: Oh yes, Internet radio. There is such a thing and it is not so well known. Basically, it is radio received over an Internet connection. Most radio stations also put their broadcasts on their website so you can go to that site and stream the audio to your PC or laptop. That means you need some kind of broadband Internet connection like a DSL line or cable TV access.

The interesting thing is that so much of what is on Internet radio is not generated by real radio stations. There are many non-radio stations that develop programs and content such as special music categories or talk sessions. I have heard estimates of 18,000 different so-called Internet radio stations.

Q: How do I receive these stations?

A: You can access individual stations on your PC if you have the speakers or headphones and know the URL. They also make special Internet radio receivers. A good example is the CC Wi-Fi radio shown in **Figure 1** from C.Crane (**www.ccrane.com**). Crane makes high end AM/FM and SW receivers. This one connects by a Wi-Fi wireless link to your home Internet router and provides a way to access all those tens of thousands of stations. No antenna required.

Q: I had not thought of buying a separate Internet radio but it may be worth it to get that many stations. Is there any other way to receive Internet radio?

A: An unexpected method is to get Internet radio on your cell phone. Apple has several apps for streaming Internet radio to their iPhones. Weather Underground (**www.wunderground.com**) recently announced their WunderRadio — an app for streaming audio not only to the iPhone but also to BlackBerrys, Windows Mobile, and any phones using Google's Android operating system.

Q: Wow, digital radio glut. Maybe I won't build that crystal set after hearing about all my digital radio options. It may be more fun to explore one of these.

A: Go modern and start checking out all your digital radio options. **NV**

The Standard for checking Capacitors in-circuit

Good enough to be the choice of Panasonic, Pioneer, NBC, ABC, Ford, JVC, NASA and thousands of independent service technicians.

Inexpensive enough to pay for itself in just one day's repairs. At $229, it's affordable.

And with a 60 day trial period, satisfaction guaranteed or money-back policy, the only thing you can lose is all the time you're currently spending on trying to repair all those dogs you've given up on.

CapAnalyzer 88A

Locate shorted or leaky components or conditions to the exact spot *in-circuit*

Still cutting up the pcb, and unsoldering every part trying to guess at where the short is?

$229

Your DVM shows the same shorted reading all along the pcb trace. LeakSeeker 82B has the resolution to find the defective component. Touch pads along the trace, and LeakSeeker beeps highest in pitch at the defect's pad. Now you can locate a shorted part only a quarter of an inch away from a good part. Short can be from 0 to 150 ohms.

LeakSeeker 82B

Available at your distributor, or call 561-487-6103

Electronic Design Specialists www.eds-inc.com

Full Speed

It writes your USB Code!

***NEW!* HIDmaker FS for Full Speed FLASH PIC18F4550**

Creates complete PC and Peripheral programs that talk to each other over USB. Ready to compile and run!

- *Large data Reports*
- *64,000 bytes/sec per Interface*
- *Easily creates devices with multiple Interfaces, even multiple Identities!*
- *Automatically does MULTITASKING*
- *Makes standard or special USB HID devices*

NEW! "Developers Guide for USB HID Peripherals" shows you how to make devices for special requirements.

Both PC and Peripheral programs *understand your data items* (even odd sized ones), and give you convenient variables to handle them.

PIC18F Compilers: PICBASIC Pro, MPASM, C18, Hi-Tech C.

PIC16C Compilers: PICBASIC Pro, MPASM, Hi-Tech C, CCS C.

PC Compilers: Delphi, C++ Builder, Visual Basic 6.

HIDmaker FS Combo: Only $599.95

Trace SYSTEMS, Inc.

DOWNLOAD the HIDmaker FS Test Drive today!

www.TraceSystemsInc.com

301-262-0300

#27 SMILEY'S WORKSHOP

C PROGRAMMING - HARDWARE - PROJECTS

AVR Memory Part 5: Bootloaders

Follow along with this series! Joe's book & kits are available at www.nutsvolts.com

by Joe Pardue

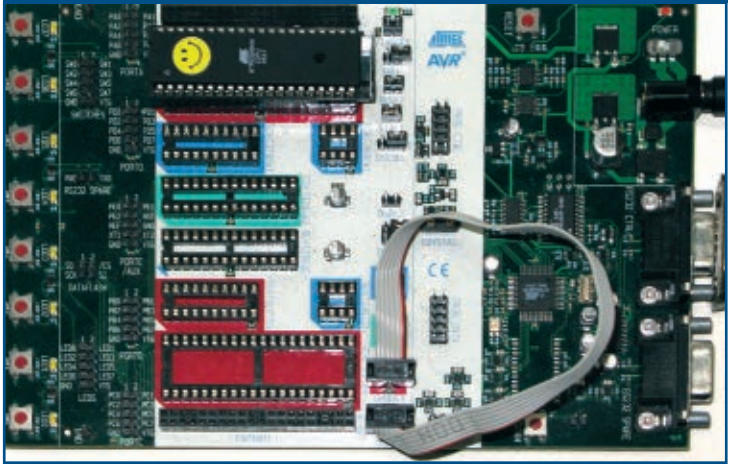
Recap

We discussed bootloaders a bit in Smiley's Workshop 22 *(Busy as a BeAVR)*, and learned about the debt they owe to Baron Munchhausen's bootstraps. This month, we'll get more practical and build on last month's AVR Memory Part 4 – Writing to AVR Flash where we learned how AVR Flash memory is structured and how to write to it. We tested that knowledge with the SmileyFlashWriter.c program. This month, we will build on this to create our own bootloader. You'll need access to last month's source code in Workshop26.zip [**NutsVolts.com** or **SmileyMicros.com**] to follow some of the instructions.

EduBootAVR - A Bootloader Written in C

A bootloader is a program that is used to download other programs. It usually resides in a special part of memory and is invoked when the device comes out of reset. It responds to data being sent — usually over a serial port — from an external device. The bootloader and the external device use a communication protocol that allows them to conduct transactions that mostly involve sending pages of binary program code to be written to the device application program section. For our demonstration, we will use avrdude on a PC to communicate with the AVR bootloader via the USART. As we will see shortly, avrdude is a very versatile tool that can handle many different programming protocols. We will be using the AVR109 protocol in avrdude but we will implement only those commands necessary for uploading program memory. There are other tasks that could be implemented such as allowing us to read and write EEPROM, but since we want to keep this bootloader small and simple, we won't add those functions.

■ FIGURE 1. STK-500 with ATmega644.



Deciding to Run the Bootloader or the Application

A bootloader is used to upload application programs. When you've uploaded an application, you then have two programs on your AVR: the application that begins at Flash memory location 0x0000 and your bootloader that begins at the NRWW (see last month) high memory location you've selected for the particular device (such as 0x3E00 for an ATmega328). Now, you have to decide which program you want to run when your AVR starts up. Generally, your system should be designed to run the application and only run the bootloader when you actually need it to upload some code. You could do this by having a function in your application that allows you to call the bootloader, but what if your application gets trashed? Then how will you upload your code? Atmel designed the AVR so that those with boot sections can have the device start up in that boot section if the BOOTRST fuse is set. With that fuse set, when the AVR starts after reset it loads the bootloader start address (also set by fuses) and starts the program from there.

The standard way to start up a microcontroller that is using a bootloader is to have the bootloader start and let it check for some indicator that it is needed. You could have the bootloader wait to see if an external programmer is trying to talk to it via the USART; after a brief pause with no attempt at contact, the bootloader can decide it isn't needed and execute a jump to the application code at 0x0000. The downside of this technique is that you want the wait to be long enough to realistically allow a bootloader communication to begin, but not so long that a user might think something has gone wrong.

worldmags

[Remember that folks using microcontrollers expect instant response and aren't used to turning them on, then getting a cup of coffee while it boots like we expect with PCs.] Usually, we would check to see if the bootloader is needed in less than a second, and then switch to the application. Another method to tell the bootloader to stay active is to designate a pin state. You could, for instance, tell it to check PORTC pin 7 and if it's high, run the boot code; if it's low, run the application. This solves the time problem since it can be done in microseconds, but it adds the cost of designating a pin in hardware that has a switch or jumper to set its state. So, each method comes with a cost; one is time, the other is extra hardware.

Use Hardware to Select the Bootloader

You could have a hardware design that has a pushbutton connected to a pin that you intend to use for the application program. There is no reason that you can't also use this to tell the bootloader to run. Just have the bootloader check that pin when it starts up out of reset. If it is pressed, then run the bootloader; if not, then run the application which can then take over the button for its own use since it is no longer needed by the bootloader. The following code snippet (which we do not use in EduBootAVR) would use a pushbutton that sets PORTD pin 7 to ground when pressed, or is pulled up to VCC when not pressed.

Place these definitions in the header:

```
#define BOOT_STATE_PIN PIND
#define BOOT_STATE_PIN_NUMBER PIND7
```

Place this code in the bootloader main() function:

```
// Use bootloader or application code?
// If BOOT_STATE_PIN_NUMBER is low,
// use bootloader.
if( !(BOOT_STATE_PIN &
(1<<BOOT_STATE_PIN_NUMBER)) )
{
    // run the bootloader code
}
else
{
    // jump to the application code
}
```

Use Software Timeout to Select the Bootloader

We want to set some kind of timeout so that when the bootloader starts up on reset, it will wait for some short period. If nothing is talking to it in a valid protocol over the USART, then it will jump to the application section. This is what we use in our bootloader. We set a count based on the clock frequency divided by 16 (this is arbitrary and for the 16 MHz board would give a maximum count of one million). Then, we check to see if we have any new USART traffic. If not, we increment the counter until it reaches the maximum. A million may seem like a lot, but remember that this puppy is making this check about every clock cycle so it does it a million times in a fraction of a second. We run this count in the receiveByte function which calls application_ptr() to jump to the application if it counts out. Otherwise, it returns the received byte.

```
uint8_t receiveByte( void )
{
  uint32_t count = 0;

  // Wait for data to be received
  while ( !(USART_CONTROL_STATUS_REG_A &
  (1<<USART_RECEIVE_COMPLETE)) )
  {
    if(count++ >= (F_CPU >> 4))
    application_ptr();
  }

  // Get and return received data from buffer
  return USART_DATA_REG;
}
```

The application_ptr() is not actually a function, but is a pointer to the first address in Flash, 0x0000, and is defined as:

```
// Function pointer to jump to the applicaion
// memory section
static void (*application_ptr)( void ) = 0x0000;
```

Okay, pointers like this are kind of weird, but don't worry, we'll discuss them eventually (if we haven't already). For now, it works, so go with it.

Other Bootloaders

There are bunches of AVR bootloaders out there, and I can't vouch for which is the best. Best is just too hard to judge. There was a long thread on **www.avrfreaks.net** [Google 'smallest bootloader site:avrfreaks.net']. On **AVRFreaks.net**, there are many bootloader projects around, but probably the two most discussed are 'blips 4 bootloader' by Steve Childress and 'fast tiny & mega uart bootloader' by Peter Danneger [Google these using site:avrfreaks.net]. There is a quite good avrfreaks thread '[FAQ][C] Bootloader FAQ' that discusses the pdf file: AVR Bootloader FAQ by Brad Schick and Cliff Lawson. I'm not providing links because avrfreaks links are very long, so you've got some Googling to do.

Please note that the bootloader we are designing for this article won't work with the Arduino using the Arduino IDE. It works just fine for the Arduino board with avrdude and the script provided here, but the Arduino IDE is hardwired to use an 'avr isp' type bootloader. If you use this bootloader with an Arduino then want to restore the original bootloader, you can get a copy at: **www.ladyada.net/library/arduino/bootloader.html**.

The AVR109 Bootloader Protocol

Atmel has promulgated several bootloader protocols — as have others — and again I can't vouch for which is the best, but I've chosen AVR109 [**www.atmel.com/atmel/acrobat/doc1644.pdf**]. This application note is accompanied by the source code written for the IAR compiler. I haven't bothered to port it to gcc since I'm only interested in a subset of the commands available. This protocol is the AVR side of the equation and the PC side is covered by the application note: AVR911 Open Source

worldmags

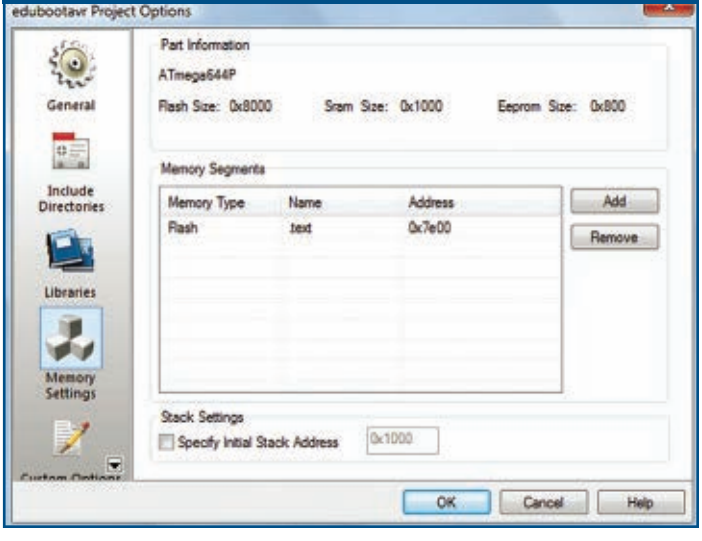
Programmer. It is compatible with the AVRProg utility that comes with AVRStudio [I use AVRProg for the Butterfly in my C Programming book, but for this Workshop we'll use avrdude which also speaks AVR109]. This protocol covers reading and writing Flash and lockbits and EEPROM and reading Fuse Bits, but we will only look at the Flash.

AVR109 Commands We Use

The heart of our bootloader is the function AVR109CommandParser() shown next. This function contains a switch statement that calls one of the AVR109 commands, depending on the command byte received over the serial port. Not all the AVR109 commands are used since only those chosen are needed for a simple bootloader. Most of the function names are the same as the command name and are somewhat self explanatory [autoIncrementAddress() is called by the command byte 'a' which tells the bootloader to auto increment the address — whoa, this code is almost self-documenting!]. [This is a joke since anyone who claims their code is self-documenting is wrong and should be ignored from that point forward (as should their code). They are also dangerous (as is their code), but we'll save that for another rant.]

```
// Note that the function name is the command
// except where otherwise commented.
void AVR109CommandParser()
{
  uint8_t cmd;

  while(1)// Loop forever
  {
    cmd = receiveByte();

    switch (cmd)
    {
      case 'P': // Enter program mode
      case 'L': // Leave program mode
        sendByte('\r');
        break;
      case 'a':
        autoIncrementAddress();
        break;
      case 'A':
        setAddress();
        break;
      case 'e':
        chipErase();
        break;
      case 'T':
        selectDeviceType();
        break;
      case 's':
        readSignatureBytes();
        break;
      case 't':
        returnSupportedDeviceCodes();
        break;
      case 'S':
        returnSoftwareIdentifier();
        break;
      case 'p':
        returnProgrammerType();
        break;
      case 'b':
        checkBlockSupport();
        break;
      case 'B': // Start Block Flash Load
      case 'g': // Start Block Flash Read
        startBlockAction(cmd);
        break;
      case 'E':
        exitBootloader();
        break;

      default:
        if(cmd != 0x1B) sendByte('?');
        break;
    }
  }
}
```

■ FIGURE 2. Project Options Custom.



Some of these functions are required by avrdude when running in the AVR109 mode, though several don't do much and wouldn't really be needed if you want to write your own PC side bootloader and use it with a device that you already know about. You wouldn't need to get the identifying info such as the signature byte, the programmer type, nor would you need to see if it supports blocks. Neither would you need to tell it to enter or leave the programming mode since what else would it do?

In that latter case, all we do is return a '\r' that lies to avrdude so that it will continue working. Why write our own when avrdude is a great and free program? I'm not going to try to write another version of it just to save a few bytes of code. Remember that this bootloader just barely gets in under the 1,024 byte limit and the next lower limit is 512 bytes, so we'd have to eliminate half the code to have a chance of getting there (and that isn't likely). If you really need to go that low, check out some of the assembly language bootloaders and PC side programs that communicate with them.

The critical action takes place in the startBlock Action(cmd) function:

```
void startBlockAction(uint8_t cmd)
{
  uint8_t tempRec;
  uint16_t tempSize;

  tempSize = (receiveByte() << 8)
  | receiveByte();
  // dummy read for type - we only do flash
  // NOT EEPROM
  tempRec = receiveByte();
  if(cmd == 'B') blockFlashLoad( tempSize );
```


worldmags

```
  else blockFlashRead( tempSize  );
}
```

This function — depending on whether cmd is 'B' or 'g' — calls functions to load or read the Flash. The load function is pretty much the same as the Flash load function we discussed last month for the SmileyFlashWriter. The blockFlashRead() function uses the avrlibc pgm_read_byte_near(address) function similar to what we saw in Workshop 25.

```
void blockFlashRead(uint16_t size)
{
       uint8_t data;
       do {
            // read_program_memory(address,0x00);
            data = pgm_read_byte_near(address++);
            // send byte
            sendByte(data);
            // reduce number of bytes to read
            // by one
               size--;


       } while (size);      // loop through size
}
```

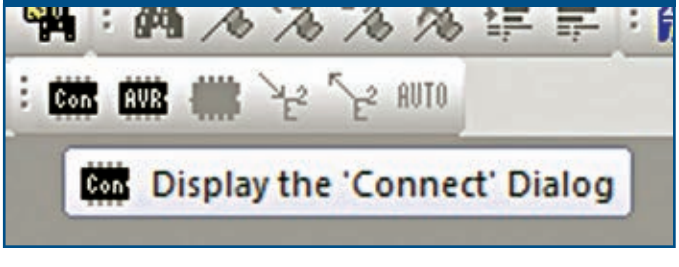
Keeping the Bootloader as Small as Possible

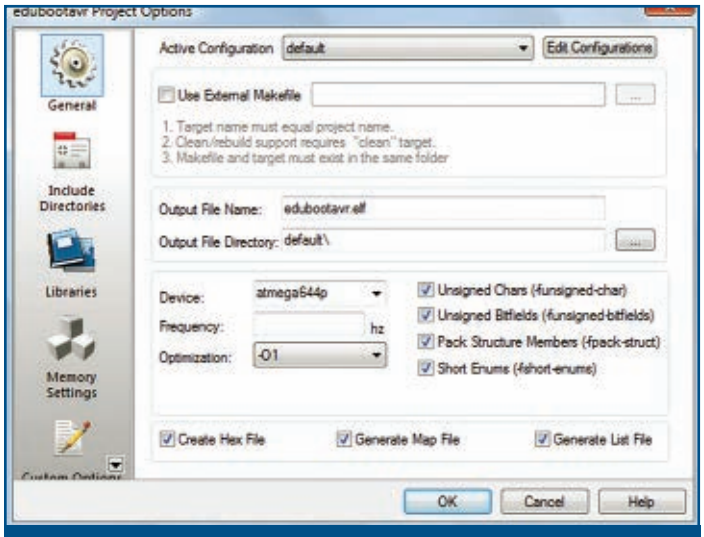
When we compile EduBootAVR with AVRStudio/WinAVR with our usual settings, we get a code size of 1,086 bytes. Oh, darn, the boot section boundary is at 1,024, so it is 62 bytes too large. Can we get the code smaller? Could we rewrite the program hoping to find something that will save us 62 piddly bytes? Well, we might, but there are other ways to save some space with a bootloader.

Eliminate Some of the Start Code

When we use AVRStudio with WinAVR to compile files, we get a gift of having necessary startup code added for us. This includes the interrupt jump table. But what if you aren't using the interrupts? Well, you can eliminate it. Since we aren't using interrupts in the bootloader, let's dump the jump table! We do this by opening the Project Configuration Options window Custom Options pane, as shown in **Figure 2**. We write '-nostartfiles' in the 'Add' test box, then click 'Add' so that '-nostartfiles' appears as shown in the **figure**. This also removes some startup code that we *do* need, so we have to add it back by putting the following code snippet right before the main() function:

■ FIGURE 4. Connect Dialog.





■ FIGURE 3. Project Options General.

```
// From Peter Fluery AVRFreaks Aug 10 2005 -
// to remove interrupt Vector table
// put -nostartfiles in LDFlags, add the
// following function saves wasted space
void __jumpMain (void) __attribute__ ((naked))
__attribute__ ((section (".init9")));

void __jumpMain(void)
{
     asm volatile ( ".set __stack, %0" :: "i"
     (RAMEND) );
     asm volatile ( "clr __zero_reg__" );
          // r1 set to 0
     asm volatile ( "rjmp main");
          // jump to main()
}
```

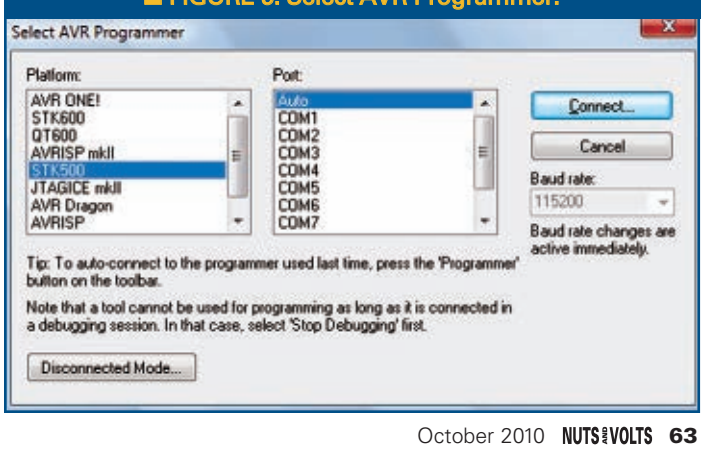
Now when we compile the code, we get 958 bytes which will fit in the desired boot section. We could stop here, but there is also another way to save some space.

Use a Different Optimization Level

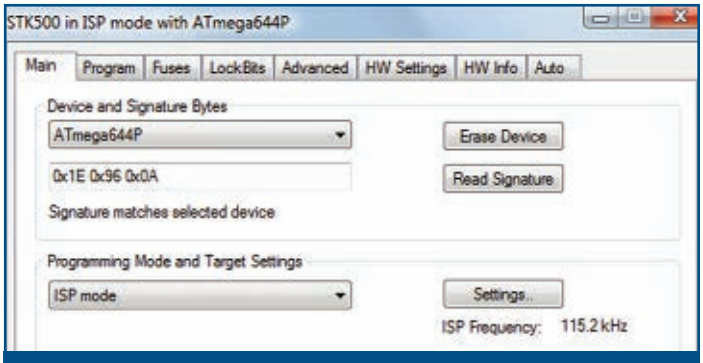
We can keep the code unchanged and instead of using the default optimization of -Os, we can use -O1. You change this as shown in **Figure 3**. Just select -O1 from the Optimizations drop box. When you compile using this optimization, you get 986 bytes which would also get you inside the limits.

If you use both, you get 854 bytes, providing you with

■ FIGURE 5. Select AVR Programmer.



worldmags



■ FIGURE 6. Programmer Main Tab.

■ FIGURE 8. Program Tab.

even further headroom. There may be even more ways to save space, but we are already down 170 bytes, so why bother? If you really want to get it smaller, check out the assembly language bootloaders mentioned earlier.

EduBootAVR Source Code

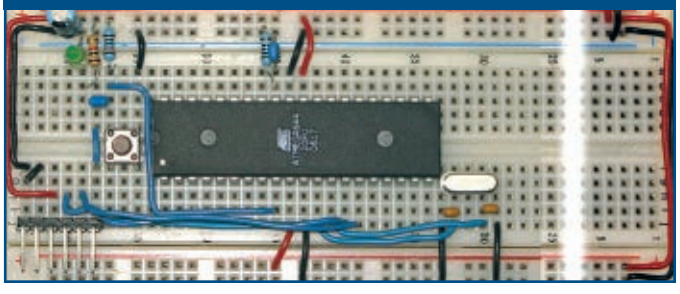
The source code for this month's article is located in Workshop27.zip as usual, and – not so usual – it is also located on Google Code Project Hosting: **https://code.google.com/p/edubootavr/source/browse/#svn/trunk**.

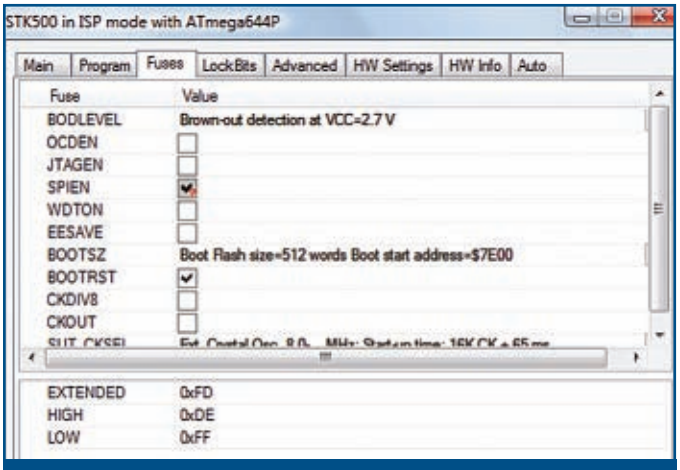
This is my first attempt to provide a Collaborative Open Project — a concept that we'll dive into next month.

Using the STK-500 to Program the ATmega644

We will use the STK-500 to program our ATmega644 chip. The setup is shown in **Figure 1**. I strongly suggest that you read AVRStudio Help\AVR Tools User Guide; click on

■ FIGURE 9. BeAVR40.





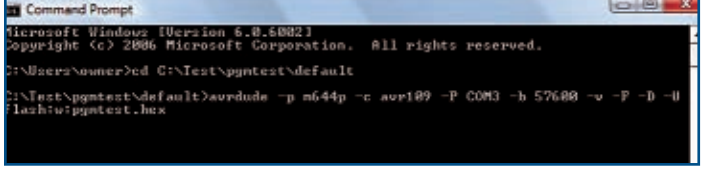
■ FIGURE 7. Fuses Tab.

the STK-500 and read about it. You'll see that there are many things that the STK-500 can do, but we'll just use it to do ISP programming on an ATmega644 chip in a socket on the board. In Workshop 22, we saw how to use the Dragon to program an ATmega644 on a breadboard (BeAVR40 design) using a hardware setup that would make Medusa look cute. If you got that working, then you can just follow the directions there and forget about the STK-500.

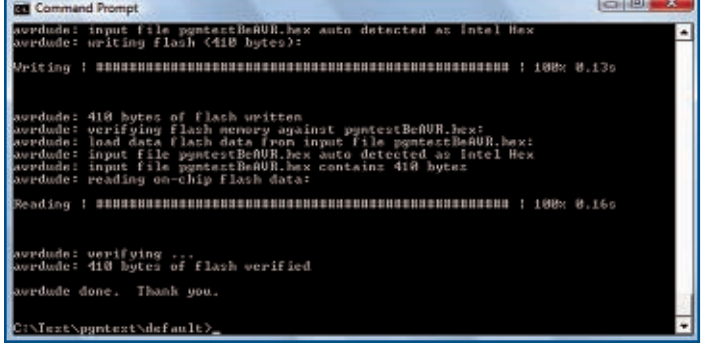
However, if you have access to an STK-500, you may have an easier time since you can plug the ATmega644 into a socket and then use a 12-pin cable that comes with the STK-500 to program the chip (thus saving Medusa for the Kraken). It is much simpler to set up and therefore less problem-prone than the other way.

In AVRStudio, click the Connect Dialog as shown in **Figure 4**. In the Select AVR Programmer window shown in **Figure 5**, highlight STK-500 and Auto, and click Connect. (If it complains that it can't find the STK-500, make sure that you are using a COM port in the acceptable range.)

■ FIGURE 10. Command Prompt.


Command Prompt
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.
C:\Users\owner>cd C:\Test\pgmtest\default
C:\Test\pgmtest\default>avrdude -p m644p -c avr109 -P COM3 -b 57600 -v -F -D -U flash:w:pgmtest.hex

■ FIGURE 11. avrdude Verifying.


Command Prompt
avrdude: input file pgmtestBeAVR.hex auto detected as Intel Hex
avrdude: writing flash (410 bytes):
Writing | 100% 0.13s
avrdude: 410 bytes of flash written
avrdude: verifying flash memory against pgmtestBeAVR.hex:
avrdude: load data flash data from input file pgmtestBeAVR.hex:
avrdude: input file pgmtestBeAVR.hex auto detected as Intel Hex
avrdude: input file pgmtestBeAVR.hex contains 410 bytes
avrdude: reading on-chip flash data:
Reading | 100% 0.16s
avrdude: verifying ...
avrdude: 410 bytes of flash verified
avrdude done. Thank you.
C:\Test\pgmtest\default>

worldmags

In the Programmer Main tab, read the signature just to verify that you really are communicating with an ATmega644 (as shown in **Figure 6**).

In the Fuses tab (see **Figure 7**), set the BOOTSZ to $7E00 and check the BOOTRST fuse.

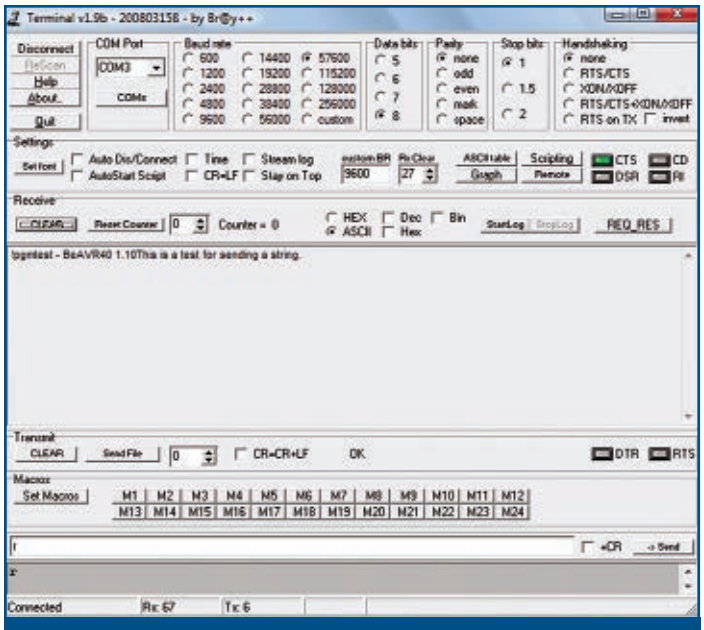
Finally (as shown in **Figure 8**), open the Program tab and make sure you are pointed to the hex file; then click Program.

Using EduBootAVR

Great! Now we have our own bootloader that we can use to upload programs. Let's take the ATmega644 that we just used the STK-500 to load the bootloader and put it on a breadboard like the BeAVR shown in **Figure 9** (discussed in detail in Smiley's Workshop 22 – Busy as a BeAVR). In earlier Workshops, we've seen how to use bootloaders with various programs. This includes AVRProg with the Butterfly; the Arduino IDE with the Arduino; and avrdude with several things. We'll finish off this month by using avrdude in the Command Prompt. We'll do this cookbook style since we've seen the details before.

- Make sure that you have the code from Workshop 25 located at: C:\pgmtest. We will upload this to our application section using edubootavr.
- Make sure that you have compiled pgmtest for the ATmega644 (BeAVR40).
- Find the Command Prompt program (probably under Start\All Programs\ Accessories\Command Prompt).
- Open Command Prompt and type in: cd C:\pgmtest\default (if that is where you've put that code) and hit return to take you to the pgmtest default directory where pgmtest.hex resides.
- Find the COMx port your BeAVR40 is talking to.
- Open NotePad and type in: avrdude -p m644p -c avr109 -P COM3 -b 57600 -v -F -D -U flash:w:pgmtestBeAVR.hex.
- Copy that and paste it to the Command Prompt as shown in **Figure 10**.
- Hit enter and it should upload pgmtest as shown in **Figure 11**.
- Open Bray's Terminal and reset the BeAVR40 which should show the text in **Figure 12**.

So, now you know all about AVR memory. If you thought this was edifying, then don't forget to check the *Nuts & Volts* WEBSTORE section in the magazine or on the Internet where you can get my books and project kits to help you learn and have some fun doing it. **NV**



■ FIGURE 12. Bray's Terminal.

Cana Kit
www.canakit.com
SAVE 15% !!!
COUPON CODE NV201002
$59.95
ASSEMBLED
New
UK1133
50A Digital PWM Motor Speed Controller
Optional LCD Module: $17.95
$29.95
Arduino
Duemilanove
$34.95
CK1301
Mini USB PIC Programmer with ZIF Socket
Assembled: $49.95
$59.95
ASSEMBLED
UK1104
4-Port USB Relay Controller with
6-Channel Analog/Digital Interface
FM Transmitters • Timers • Audio Amplifiers • Motor Controllers • Custom Designs
www.canakit.com
1•888•540-KITS (5487)

worldmags

ELECTRONET

www.canakit.com

- Over 200 Kits & Modules to choose from!
- Easy Worldwide Online Ordering
- Fast Delivery & Same Day Shipping

FM Transmitters • Timers • Audio Amplifiers • Motor Controllers

Pololu
Robotics & Electronics
WWW.POLOLU.COM

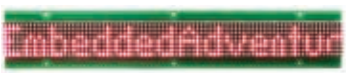
OBD2 Adapter kit for only $35

www.obddiag.net

ramsey
www.ramseykits.com
AM/FM Broadcasters • Hobby Kits
Learnng Kits • Test Equipment
...AND LOTS OF NEAT STUFF!

embedded adventures
www.embeddedadventures.com

With tutorials, project ideas, modules, kits, LED panels and more



GPS MADE SIMPLE™

LinxTechnologies.com
LOCATE • TRACK • MAP • FIND • NAVIGATE

USB Add USB to your next project— it's easier than you might think!

DLP Design

• USB-FIFO • USB-UART • USB/Microcontroller boards
• RFID Reader/Writer • ZigBee Ready Transceivers
www.dlpdesign.com
Absolutely NO driver software development required!

Sheet Metal Fabrication

Complete Fabrication Center
Integrated Ideas & Technologies, Inc.
Precision Laser, Waterjet, Plasma, Machining, Forming, and Welding Capabilities
www.iitmetalfab.com

PAiA
paia.com
Electronic Music Kits and More

Music/Tutorials/Kits
Analog Synthesizers
Guitar Effects
Tube Electronics
Studio Gear

ServoCenter
servo-center.com
· 16 servos, 16 I/O, 8 A/D
· USB, RS-232, TTL serial
· 14-bit servo control
· Built-in SC-BASIC Sequencer

THE ORIGINAL SINCE 1996
PCB-POOL®
Beta LAYOUT

- Low Cost PCB prototypes
- Free laser SMT stencil with all Proto orders

WWW.PCB-POOL.COM

ALL ELECTRONICS CORPORATION
Electronic Parts & Supplies Since 1967

GOT HAT?
www.nutsvolts.com

Das Blinkenboard
Not just for blinken LEDs.
Much Much More!
Complete kits available @
http://store.nutsvolts.com & http://store.servomagazine.com

For the ElectroNet online, go to
www.nutsvolts.com
click Electro-Net

BY RON HACKETT

IMPLEMENTING AN ADC KEYPAD

Way back in March '07, well before the PICAXE Primer appeared on the scene, I wrote an article for *Nuts & Volts* that described how to interface a 12- or 16-key matrix keypad to a PICAXE-18X processor. In the article, I used a standard matrix-scanning approach to decoding a keypress. For a 4 x 4 matrix keypad, this approach requires eight I/O lines (four inputs and four outputs) for the interface. That's a large number of I/O lines to dedicate to a single function, but I didn't mind the I/O "expense" because the keypad was being used as part of a stand-alone I/O terminal and the 18X processor had more than sufficient resources for the task. However, I'm sure you can think of several 08M-based projects that would also benefit from user input via a keypad, so this month we're going to focus on a different approach to interfacing a matrix keypad with a PICAXE processor. This alternate approach only requires a single ADC input to decode a 12- or 16-key matrix keypad, so it's suitable for use with any 08M-based project. Of course, the same approach can be used with any PICAXE processor, as well.

OLD BUSINESS

Before we get into the details of this month's project, I have a small piece of "old business" to discuss. Back in the April '10 installment of the Primer (the one in which we developed software for our MAX7219 LED Display Driver), I mentioned a problem I had in getting the driver software to function correctly. (See the "Testing the Serialized LED Display" section that starts on page 18 of that article.) I initially tried to "blank" the four LED digits by using the following code snippet:

```
for maxreg = 1 to 4
  outbyte = blank
  gosub shout
next maxreg
```

The problem was that this approach didn't work, but I couldn't figure out why, and my publication deadline was rapidly approaching. To "solve" the problem, I took the easy way out and modified the code so that it did work, but I did invite readers to figure out what was going on.

Fortunately, I received an email from an astute reader (Matt Rusnak) that clarified the cause of my little problem — thanks Matt! In effect, I had ignored my own advice.

In the previous Primer column (February '10), I reminded readers that whenever a word variable is declared (e.g., *symbol outword = w0*), it's important to not also declare either of the associated byte variables (e.g., b0 or b1) because changing the value of one variable (e.g., b1) will inadvertently also change the value of another variable (w0, in this case).

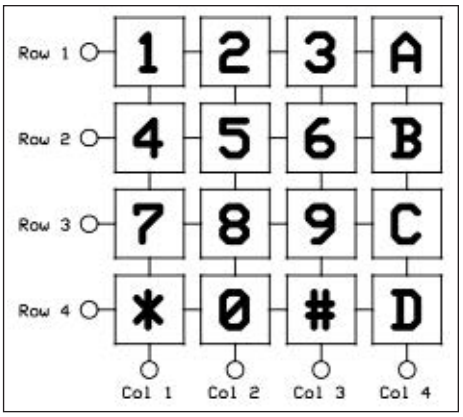
I then went on to explain why our MAX7219 project was an exception to this rule (see the original article for the explanation), and to make the following declarations in the driver software:

```
symbol outbyte = b0
' data to be transmitted
' to the MAX7219

symbol maxreg  = b1
' MAX7219 register that
' receives data

symbol outword = w1
' concatenation of maxreg
' and outbyte
```

So far, so good, but the problem entered the picture when I forgot the fact that the *shout* subroutine modifies *outword* (w0). Therefore, it also modifies *maxreg* (as well as *outbyte*, but that's not a problem), which causes the above *for-next* loop to misbehave.

As usual, once the cause of the problem is clearly understood, the solution is relatively simple. All we need to do is declare an additional byte variable to use in the *for-next* loop. (Of course, we can't use *index*, because the *shout* subroutine uses that variable as well.) It's a little embarrassing to have been "hoist by my own petard," as they say, but I do appreciate the feedback and thought it worthwhile to pass the information on to everyone. It certainly emphasizes the need for caution

worldmags

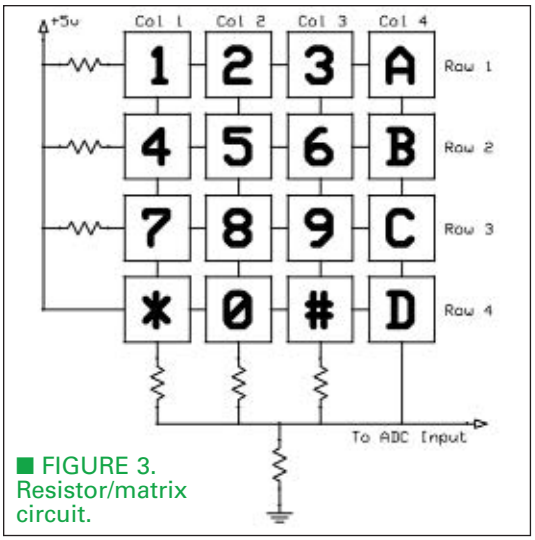


■ FIGURE 1. Schematic of 4 x 4 matrix keypad.

*Pin	Connection
1	Row 2
2	Row 3
3	Column 1
4	Row 4
5	Column 2
6	Column 3
7	Column 4
8	Row 1

* From left to right

■ FIGURE 2. Pinout for 4 x 4 matrix keypad.



■ FIGURE 3. Resistor/matrix circuit.

when declaring "overlapping" word and byte variables.

NEW BUSINESS

There's another bit of information I want to share with everyone. I'm pleased to announce the recent publication of my first book, *PICAXE® Microcontroller Projects for the Evil Genius*. By the time you read this, it will be available at the *NV* Webstore (**http://store.nutsvolts.com**). In addition to simply being a shameless plug for the book, I'm also mentioning it here because I need to include the following statement: Portions of the following project are excerpted from the book with permission from McGraw-Hill.

INTERFACING A 4 X 4 MATRIX KEYPAD

Actually, the *Evil Genius* book includes three different projects that involve a matrix keypad. We're going to be focusing on a variation of the first project, but when we have finished I'll also include some information on what we did next in the book. We'll be using the same keypad from the book's projects (a 4 x 4 matrix), but you certainly can adapt the project for use with a 4 x 3 matrix keypad if you prefer.

The standard arrangement for a 4 x 4 matrix keypad is shown in **Figure 1**. Whenever a key is pressed, the connections for the corresponding row and column are shorted together; for example, pressing the "6" key connects the pins for row 2 and column 3. You don't need to use the same keypad that I did. Any matrix keypad you have on hand should work just as well — just be sure that it is in fact a matrix layout. I have seen keypads that look similar to a "genuine" matrix, but actually have each key connected to its own line with a common connection to all the keys. It would be possible to use this type of keypad, but the necessary modifications would be extensive.

The keypad that I am using includes a row of eight holes along its bottom edge, into which a male header can be soldered so that the keypad can easily be connected to a breadboard, stripboard, or ribbon cable assembly. For breadboard projects, it would actually be more convenient to have the header at the top so that the keypad could be placed in front of the breadboard rather than behind it. If you prefer that arrangement, you may be interested in the second *Evil Genius* keypad project in which we perform a little "surgery" on the keypad (more about that later). For now, we'll stick with the stock configuration.

There seems to be a fair amount of variation in the specific order of the connecting points on different matrix keypads. The simplest arrangement for the user would naturally be to have the row and column connections in order. However, probably because the keypad layout is simpler and therefore less expensive, the connections are frequently not in any logical order. **Figure 2** shows the pin assignments for the keypad that I used (which is available on my website at **www.jrhackett.net**). If you use a keypad that has a different pin arrangement, you will need to modify the breadboard layout. The schematic, of course, will remain the same.

USING AN ADC APPROACH TO DECODING MATRIX KEYPADS

The ADC approach to decoding a matrix keypad employs a resistor matrix to produce an analog voltage level that differs for each possible key-press. (The output voltage remains at ground level as long as no key is pressed.) The analog voltage is connected to an ADC input and the processor simply converts the resulting voltage level back to the corresponding keypress. **Figure 3** presents one possible arrangement for a resistor matrix that can be used in the ADC approach to keypad decoding.

As you can see, each possible key-press connects one, two, or three resistors in series between the +5V line and ground. By carefully choosing the values of the seven resistors in the layout, it's possible to produce 16 voltage levels that are spread out far enough to easily differentiate. I would like to be able to say that I used a sophisticated mathematical formula to determine the required resistor values, but I actually used a more primitive "trial

and error" approach to the problem.

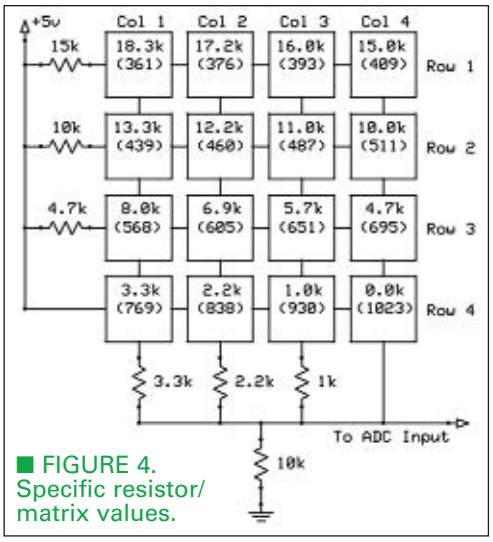
Before I discuss my method and the results I obtained, I need to mention an important factor to keep in mind. Standard 1/4 watt resistors have a 5% tolerance rating which means that a 10K resistor can actually measure anywhere between 9.5K and 10.5K. This is why it's important to be able to produce a wide range of analog voltages. If the ADC readings for two adjacent keys were too close to each other, variations in actual resistor values could result in misidentifying the specific key that has been pressed.

In order to make my trial and error approach as painless as possible, I set up a simple Excel spreadsheet to compute the ADC values that would result from a specific combination of resistors and then tried various combinations until I found one that worked. When I used the 256 levels provided by the *readadc* command, some of the ADC values for two adjacent keys were so close (differences of 4 or 5) that I was concerned that errors could result. Switching to the 1024 levels provided by the *readadc10* command greatly simplified the task.

The resistor values that I finally chose are shown in **Figure 4**. Each of the 16 "key" positions includes two pieces of relevant data: the total resistance that is connected in series with the 10K base resistor when the corresponding key is pressed; and (in parentheses) the resulting value produced by the *readadc10* command.

If you would like to experiment with different resistor values, the formula you need is the basic voltage divider rule. In English: The ADC reading is to the maximum ADC value (1023 for *readadc10*) as the base resistance (10K) is to the total resistance. **Figure 5** presents the same thing algebraically if you prefer it that way.

If you double-check some of my computations, you'll find that they are sometimes off by a small amount. That's because I didn't round anything up; I truncated all my results because that's what the 08M and all PICAXE processors do. Finally, I need to emphasize that these are theoretical results; your specific ADC values will almost certainly be somewhat different. We'll confront that issue in the next section when we actually construct and test our breadboard circuit.



■ FIGURE 4. Specific resistor/ matrix values.

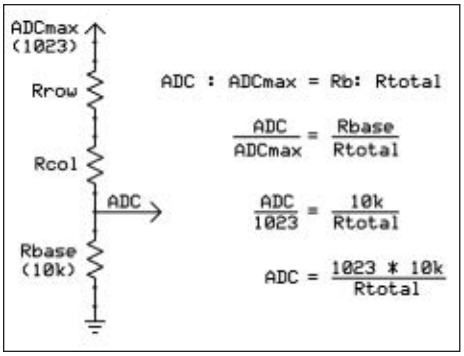
CONSTRUCTING AND TESTING THE BREADBOARD CIRCUIT

The schematic for our breadboard circuit is shown in **Figure 6**. As I mentioned earlier, the eight connections to the keyboard that I'm using are not logically ordered, but it really doesn't matter much. The important thing is to make sure that the connections for each resistor are the same as the ones presented earlier in **Figure 2**. If the pinout is different for the keypad you intend to use, simply rearrange the connections appropriately. The parts list for our breadboard circuit is too simple to warrant a separate sidebar; just a matrix keypad with a male header, the seven resistors shown in the schematic, and a PICAXE-08M processor are needed.

My breadboard layout is shown in **Figure 7**. To save some space on the breadboard, I'm using the AxMate-FT programming adapter which is also supplying power to the breadboard. The small red printed circuit board (PCB) attached to the AxMate-FT is the 5V version of SparkFun's latest FTDI Basic Breakout board (DEV-09716) which I have reconfigured to work correctly with PICAXE processors (as we discussed back in the June Primer). I really like SparkFun's new board because they have moved the six-pin female header underneath which makes this the smallest FTDI-based board that I have found so far.

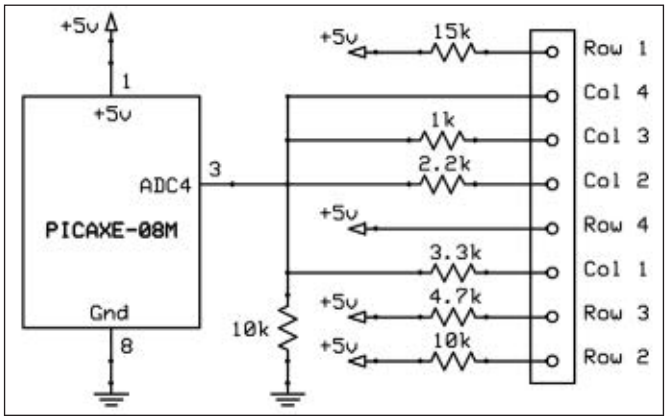
When I assembled the AxMate-FT in the photo, I installed its six-pin male header on the bottom of the board so that the two boards line up correctly. (If you're not using an AxMate-FT adapter, you will need to install all the parts farther to the right on the breadboard to allow enough room for a power supply and programming adapter.)

When you have assembled your breadboard circuit, we're ready to discuss our first program (*keypad1.bas*). Download it from the *Nuts & Volts* website (**www.nutsvolts.com**), along with *keypad2.bas* which we'll discuss

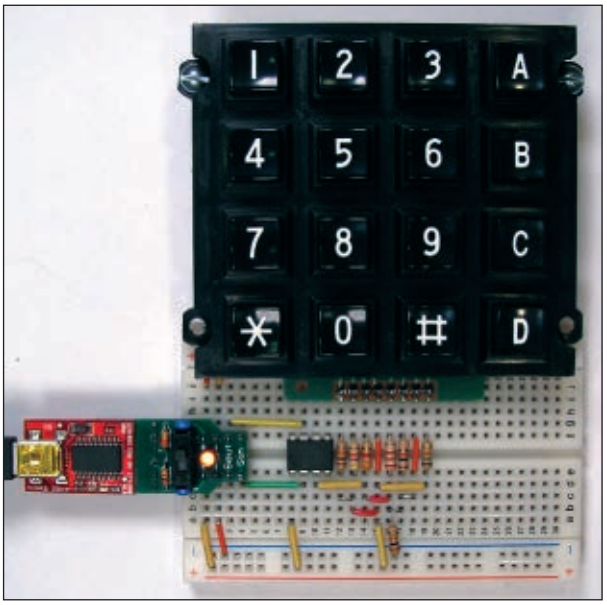


■ FIGURE 5. The voltage divider formula.

■ FIGURE 6. Schematic for ADC matrix keypad circuit.



worldmags



■ FIGURE 7. Breadboard layout for ADC matrix keypad circuit.

■ FIGURE 8. Matrix keypad after "surgery."

shortly. (At this point, you may want to print out both programs for reference throughout the following discussion.) *Keypad1.bas* has two purposes: to make sure that the breadboard circuit is configured correctly, and to determine how closely our "real-world" ADC values match the theoretical values that were presented back in **Figure 4**. The main program loop repetitively carries out the following tasks:

- Wait for keypress. (Theoretically *junk* = 0 if no key is pressed, but "< 5" is safer.)
- Pause for 50 mS to de-bounce the switch. (The *pause 100* is halved at 8 MHz.)
- Get ADC value (using *readadc10*).
- Wait until switch is released.
- Send the digits of the value to the terminal window (followed by *CR* and *LF*).

When you have downloaded *keypad1.bas* to your 08M, pressing any key should produce a value between 0 and 1023 in the terminal window. If it doesn't, you'll need to trouble-shoot your breadboard setup. When your circuit is functioning properly, jot down the value you obtain for each keypress. We're going to use these results soon, when we get to the keypad2.bas program. Be sure to press each key a few times because you may get slightly variable readings for some of the keys. If so, just choose the most typical value for each key and make a note of it. (As I mentioned earlier, your results will most likely differ somewhat from the theoretical values back in **Figure 4**, but they should be close.)

DECODING THE KEYPRESSES

Now that you know the ADC value that's produced by each keypress in your breadboard setup, we need to modify *keypad1.bas* so that it actually decodes each keypress and outputs the appropriate character. Naturally, we'll need another variable in which to store the resulting character; let's call it *char*. Also, we're going to use a *select case* statement to accomplish the decoding, but we can't use a series of equalities to convert each ADC value to the correct character because (as we just discovered) some of the values are slightly variable. Additional slight variation can be introduced by changes in temperature and other factors. To make sure that we always decode the correct character, we'll use a series of "less than" phrases in the *select case* statement and work our way up the list. Using this approach, our *select case* statement will take the form of the following code fragment:

```
select case key
  case < ?
    char = 49   ' ASCII code
                ' for "1"
  case < ?
    char = 50   ' ASCII code
                ' for "2"
  ' etc., etc.
```

The question remains: What specific values should we use in place of the question marks? The safest (i.e., most error-free) choice is the mid-point between the values for each pair of adjacent keys. I'll use our theoretical values to clarify this point, but you should substitute the actual values you obtained from running the *keypad1.bas* program. The theoretical value associated with the "1" key is 361 and the value associated with "2" is 376, so the mid-point is about 369. Therefore, in our *select case* statement, we'll say that any value less than 369 will be decoded as the "1" character. If we use the same approach to each of the characters, our *select case* statement becomes:

```
select case key
  case < 369
    char = 49   ' ASCII code
                ' for "1"
  case < 385
    char = 50   ' ASCII code
                ' for "2"
  ' etc., etc.
```

Since our *select case* statement involves 16 distinct cases, it will be fairly long. To compress it a bit, we're going to use a little shortcut. Similarly to many dialects of BASIC, PICAXE BASIC supports the use of the ":" symbol to separate multiple statements on the same line which means we can write:

```
select case key
  case < 369 : char = 49
       ' ASCII code for "1"
  case < 385 : char = 50
       ' ASCII code for "2"
  case < 402 : char = 51
       ' ASCII code for "3"
  case < 434 : char = 65
       ' ASCII code for "A"
  ' etc., etc.
```

The ":" shortcut is not something I'm suggesting you use frequently in your programs; it can easily make code much more difficult to read. However, our long *select case* statement is perfectly readable in this form, and also much shorter. One final point – don't forget, as soon as one of the *case* conditions evaluates to true, the associated code is executed and the remainder of the *select case* statement is skipped. For example, if *key* equals 375, *char* is set equal to 50 and the compiler jumps ahead to the program line that follows the *endselect* statement.

worldmags

Our second program (*keypad2.bas*) incorporates all the changes we just discussed. Open it in the Programming Editor and change the theoretical mid-point values in each of the *case* conditions to the values you calculated from the results of running *keypad1.bas*. Also note that the *sertxd* statement has changed; instead of sending the individual digits of the ADC value, we're now transmitting the appropriate ASCII value for each character to the terminal window. Download the program to your breadboard circuit and test all the keypresses. You should see the correct character appear in the terminal window in response to each keypress. If not, a little trouble-shooting is in order.

WHAT'S AN EVIL GENIUS TO DO?

That's as far as we're going to take our ADC keypad project this month. I'm sure it's no surprise that the project we tackled in the *Evil Genius* book was to design a serialized version that simplified the interface with a master processor by freeing the master from the chore of repetitively monitoring an ADC input. Next, we performed a little "surgery" on our keypad. We disassembled it and rearranged its innards, so that the connector was moved to the top of the keypad and its keys were shuffled into a more convenient pattern. The results of our little operation are shown in **Figure 8**.

As you can see, it looks like four of the keys have been replaced, but actually the arrows were printed on a Brother PT-300 label maker and glued onto the four keys. I used black tape with white letters so that I could completely cover the original characters on the keys. Also, I deliberately retained the "B" key as a "Back" key in a menu structure, and the "A" key as an "Accept" ("Enter") key. I realize the new key layout results in a slightly unusual arrangement for the 10 digit keys, but I think this compromise is justified by the convenient layout for the four arrow keys and the fact that the "Enter" key is in the lower right-hand corner where it should be.

Next, the modified keypad was used to develop a programmable, multi-function peripheral device (MPD) that can be easily connected to any breadboard circuit (see **Figure 9**). Because the MPD is fully programmable, it can be used to implement a variety of helpful utility and troubleshooting applications for use with any PICAXE project. The following are just a few of the many possibilities:

- Stand-alone I/O terminal with sound
- Digital frequency generator
- Frequency measurement of project signals
- Timing of project routines and waveforms

Because the MPD is based on a PICAXE-20X2 circuit, it can implement any of these functions with a surprising amount of speed and accuracy. In addition, the 20X2 has enough memory to store several programs at once. In fact, we also developed a simple operating system (MPDOS) that makes it easy to switch from program to program as the need arises.

COMING ATTRACTIONS

In addition to the publication of *PICAXE® Microcontroller Projects for the Evil Genius*, there's another major event that's about to occur in the PICAXE world. Revolution Education has announced its newest processor — the PICAXE-18M2 — and it's scheduled to be available by the time you're reading this column. The 18M2 is a revolutionary product with many impressive hardware and software enhancements. I don't have space this month to get into the details, but take a look at **Figure 10**. I think you'll be impressed.

The 18M2 also includes several software enhancements; I'll just mention two of them to whet your appetite. There's a new built-in *time* variable that keeps track of elapsed time in the background while your program is carrying out other tasks. This means that a project will be able to include a real-time clock feature without necessitating additional hardware or complicated software interrupt routines. Also, the 18M2 will be able to run up to four different program tasks in parallel at the same time by rapidly switching among the various program "threads." This new capability isn't multitasking in the strict sense, because program instructions are still being executed one at a time. However, the automatic switching among program threads will occur so rapidly that it will create the illusion of multitasking.

I have saved the most impressive specification of the new 18M2 processor until last; it's slated to retail for about the same price as the older 18M processor. To be able to purchase that amount of processing power for less than $4 is truly amazing. I'm eager to get my hands on one, and as soon as I do (which will almost certainly be in time for the next Primer column), I'll get to work on reporting what I learn about this impressive new chip. See you then ... **NV**

■ FIGURE 9. The Evil Genius MPD.

	18M2	18M
Program Storage (bytes)	2048	256
General Purpose Vars (bytes)	28	14
Storage Variables (bytes)	256	48
EEPROM Storage (bytes)	256	256-prog
Max Operating Speed (MHz)	32	8
Min Operating Voltage (V)	1.8	3.0

■ FIGURE 10. Comparison of 18M2 vs. 18M selected features.

worldmags

For a complete product detail visit our webstore!!

The Nuts & Volts WEBSTORE

ELECTRONICS

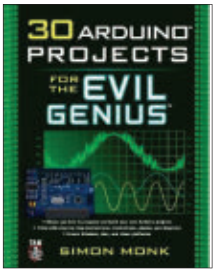
TEARDOWNS
Learn How Electronics Work by Taking Them Apart
by Bryan Bergeron

"The text is written as if Dr. Bergeron, who is a highly experienced electronics practitioner, is speaking directly to the reader with a point-by-point commentary about each teardown, complete with clear explanations of the operation and function of every component. By the time the product is completely disassembled, the reader understands the design tricks, component selection, and packaging choices that enabled the product to reach the market."
From the Foreword by Forrest M. Mims III

Amp up your knowledge of electronics by deconstructing common devices and analyzing the revealed components and circuitry. *Teardowns: Learn How Electronics Work by Taking Them Apart* contains 14 projects that expose the inner workings of household appliances, workbench measuring instruments, and musical equipment. Discover how resistors, capacitors, sensors, transducers, and transistors function in real circuitry.
Reg Price $24.95 **Sale Price $21.95**

NEW!

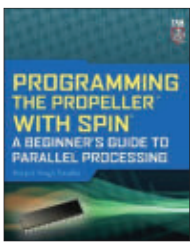
30 Arduino Projects for the Evil Genius
by Simon Monk
30 Ways to Have Some Computer-Controlled Evil Fun!



Using easy-to-find components and equipment, this do-it-yourself book explains how to attach an Arduino board to your computer, program it, and connect electronics to it to create fiendishly fun projects. The only limit is your imagination!
$24.95

We also have 30 other Evil Genius titles in stock!

Programming the Propeller with Spin
by Harprit Singh Sandhu



Programming the Propeller with Spin: A Beginner's Guide to Parallel Processing walks you through the essential skills you need to build and control devices using the Propeller chip and its parallel processing environment. Find out how to use each of the identical 32-bit processors — known as cogs — and make the eight cogs effectively interact with each other. The book covers Propeller hardware and software setup, memory, and the Spin language.
$29.95

NEW!

PICAXE Microcontroller Projects for the Evil Genius
by Ron Hackett
WHIP UP SOME FIENDISHLY FUN PICAXE DEVICES

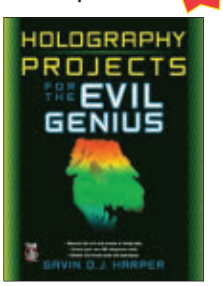
This wickedly inventive guide shows you how to program, build, and debug a variety of PICAXE microcontroller projects. *PICAXE Microcontroller Projects for the Evil Genius* gets you started with programming and I/O interfacing right away, and then shows you how to develop a master processor circuit.
$24.95

TRON.IX Book 1
by Gary Gibson
Hands-On Electronics Experiments!

This is the beautiful, fully-illustrated manual for a hands-on lab to help you build 40 of the most interesting experiments you have ever seen on a solderless circuit board, and as you do, you learn how the basic components work in a circuit. You will enjoy hours of fun creating all kinds of sounds and light displays.
$14.95

Holography Projects for the Evil Genius
by Gavin D J Harper

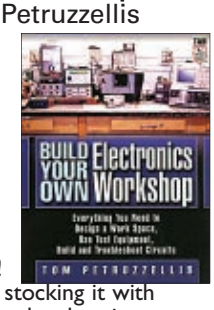
NEW!



New in the best selling Evil Genius series, this is a step-by-step guide to the improved art and science of holography, complete with explanations of the techniques, materials, processes, and tools needed to create impressive holograms. The book features more than a dozen distinct projects with more than 100 holograms posted on an associated website.
$24.95

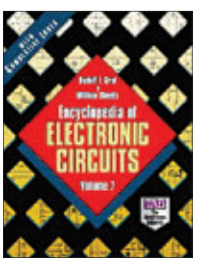
Build Your Own Electronics Workshop
by Thomas Petruzzellis

BUILD YOUR OWN DREAM ELECTRONICS LAB!



This value-packed resource provides everything needed to put together a fully functioning home electronics workshop! From finding space to stocking it with components, to putting the shop into action — building, testing, and troubleshooting systems. The best part is, this book will save you money.
Reg Price $29.95
Sale Price $24.95

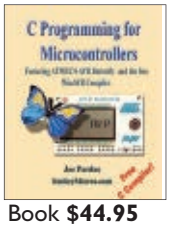
Encyclopedia of Electronic Circuits, Volume 7
by Rudolf F. Graf / William Sheets



Designed for quick reference and on-the-job use, the *Encyclopedia of Electronic Circuits, Vol 7*, puts over 1,000 state-of-the-art electronic and integrated circuit designs at your fingertips. Organized alphabetically by circuit type, this all-new collection includes the latest designs from industry giants such as Advanced Micro Devices, Motorola, Teledyne, General Electric, and others.
$39.95

Order online @ **www.nutsvolts.com**
Or CALL **1-800-783-4624** today!

BOOK & KIT COMBOS



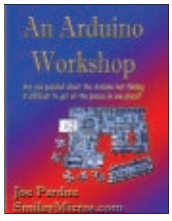
Book **$44.95**

From the Smiley Workshop
C Programming for Microcontrollers
by Joe Pardue

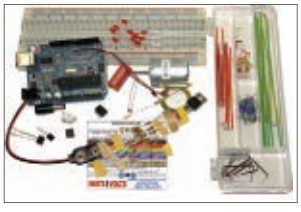
Kit **$66.95**

Do you want a low cost way to learn C programming for microcontrollers? This 300 page book and software CD show you how to use ATMEL's AVR Butterfly board and the FREE WinAVR C compiler to make a very inexpensive system for using C to develop microcontroller projects.

Combo Price $99.95 Plus S/H



From the Smiley Workshop
An Arduino Workshop
by Joe Pardue



The book *An Arduino Workshop* and the associated hardware projects kit bring all the pieces of the puzzle together in one place. With this, you will learn to: blink eight LEDs (Cylon Eyes); read a pushbutton and 8-bit DIP switch; sense voltage, light, and temperature; make music on a piezo element; sense edges and gray levels; optically isolate voltages; fade an LED with PWM; control motor speed; and more!

An Arduino Workshop Combo

Reg Price $ 124.95 **Subscriber's Price $119.95 Plus S/H**

CD ROM SPECIALS

Reg Price $79.95 **Subscriber's Price $75.95**

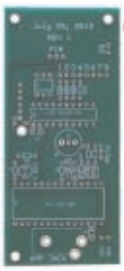
This lab — from the good people at GSS Tech Ed — will show you 40 of the most simple and interesting experiments and lessons you have ever seen on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit. Along with the purchase of the lab, you will receive a special password to access the fantastic online interactive software to help you fully understand all the electronic principles. For a complete product description and sample software, please visit our webstore.

worldmags

CALL 1-800-783-4624 today!
Or Order online @ www.nutsvolts.com

PROJECTS

Shazam!



As seen on the October 2010 cover

Halloween is one of America's favorite holidays. This project was designed to be flexible and reusable for other holidays. With a few of your own components added to the PCB and the preprogrammed chip, you'll be ready to design your own scary holiday prop.

Includes an article reprint.

Subscriber's Price **$28.45**

Non-Subscriber's Price **$30.95**

Prop Dropper Kit

If you're looking for a fun and easy way to "Drop-a-Prop," here it is! It can easily be modified to drop any prop for any occasion.

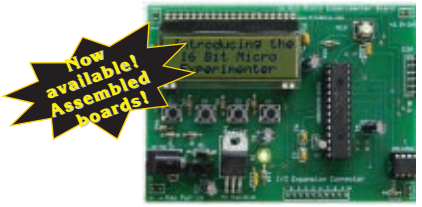
The kit includes everything you need except for a couple items of your choice, prop, elastic string, enclosure & a battery. For more information, please see the October 2009 issue.

Includes an article reprint.

Reg Price $84.95

Sale Price $63.95

16-Bit Micro Experimenter Board



Ready to move on from 8-bit to 16-bit microcontrollers? Well, you're in luck! In the December 2009 *Nuts & Volts* issue, you're introduced to "the 16-Bit Micro Experimenter." The kit comes with a CD-ROM that contains details on assembly, operation, as well as an assortment of ready-made applications. New applications will be added in upcoming months.

Subscriber's Price **$55.95**

Non-Subscriber's Price **$59.95**

Talking Skull Kit

Get ready for a tech-scary Halloween!

New for 2010, the audio board is now assembled!

For more information, please visit http://store.nutsvolts.com

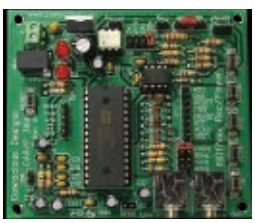
Subscriber's Price **$69.95**

Non-Subscriber's Price **$73.75**

Fully assembled kits also available!

CAR P300-120 Audio Board Kit

The optional CAR P300-120 audio board can be added to this kit. See Talking Skull Updates in *Nuts & Volts'* September 2009 issue (pg 87).



Subscriber's Price **$37.95**

Non-Subscriber's Price **$39.75**

Das Blinkenboard Kit

This kit includes a pre-programmed ATtiny84 microcontroller that sports eight software PWM channels to control motor speed and light brightness. Jumper selectable patterns can be used to operate motors, solenoid valves, relays.

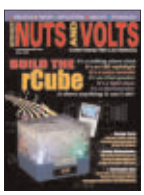
Expand your board with the "Der Magnetfelder Detektor" component pack.

Subscriber's Price **$32.45**

Non-Subscriber's Price **$35.95**

rCube Talking Alarm Clock Kit

As seen on the May 2009 cover



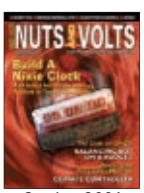
Available in blue, black, red, and green. All components are pre-cut and pre-bent for easier assembly and the microcontrollers are pre-programmed with the software. Kits also include PCB, AC adapter, and instructions on CD-ROM.

Subscriber's Price **$49.95**

Non-Subscriber's Price **$54.00**

Nixie Tube Clock Kit

Cherry Wood Clock Case



October 2006

Nixie tube clocks fuse the spirit, drama, and eerie beauty of cold war technology with modern inner works to create uncommon handcrafted timepieces.

Now with optional case choices!

Get more info @ our webstore.

Kit includes article reprint, complete instructions, and parts list

Subscriber's Price **$146.95**

Non-Subscriber's Price **$159.95**

NixieNeon Clock Kit

Using Neon and Nixie bulbs, this clock displays the precision movement of time and is programmable to display "Time Chaos" at different intervals. The NixieNeon Clock then resets itself to the correct time in an eye-catching ballet of luminosity. The PCB is 7.25"x 7.25" and consists of over 400 components.

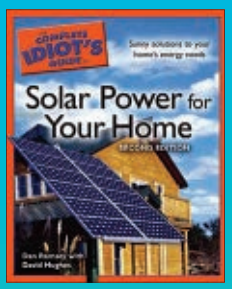
Reg Price $195.95 **Sale Price $159.95**

WE ACCEPT VISA, MC, AMEX, and DISCOVER. Prices do not include shipping and may be subject to change.

The Nuts & Volts **WEBSTORE**

ALTERNATIVE ENERGY SECTION

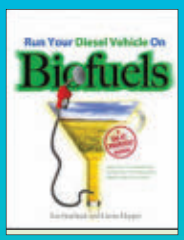
The Complete Idiot's Guide to Solar Power for Your Home
by Dan Ramsey / David Hughes



The perfect source for solar power — fully illustrated. This book helps readers understand the basics of solar power and other renewable energy sources, explore whether solar power makes sense for them, what their options are, and what's involved with installing various on- and off-grid systems.
$19.95

Do you know how many watts (YOUR MONEY) are going down the drain from "THE PHANTOM DRAW?"
If you are interested in your own power usage we at *Nuts & Volts Magazine* believe that this is the best way to help you determine your electrical energy use in ON and OFF home appliances.
To order call 1 800 783-4624
or online www.nutsvolts.com
$29.95 plus S&H

Run Your Diesel Vehicle on Biofuels A Do-It-Yourself Manual
by Jon Starbuck, Gavin D J Harper

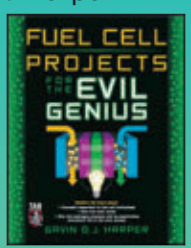


CONVERT TO BIODIESEL FOR A MORE ENVIRONMENTALLY FRIENDLY RIDE! *Run Your Diesel Vehicle on Biofuels* has everything you need to make the switch from expensive, environment-damaging carbon fuel to cheap (and, in many cases, free), clean fuel for your vehicle. Practical and decidedly apolitical, this unique guide focuses on technical details, parts, and instructions.
$24.95

50 Green Projects for the Evil Genius
by Jamil Shariff

Using easy-to-find parts and tools, this do-it-yourself guide offers a wide variety of environmentally focused projects you can accomplish on your own. Topics covered include transportation, alternative fuels, solar, wind, and hydro power, home insulation, construction, and more. The projects in this unique guide range from easy to more complex and are designed to optimize your time and simplify your life!
$24.95

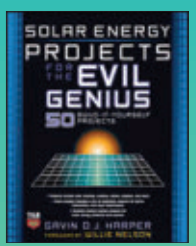
Fuel Cell Projects for the Evil Genius
by Gavin D J Harper



FUEL YOUR EVIL URGES WHILE YOU BUILD GREEN ENERGY PROJECTS!

Go green as you amass power! Fuel Cell Projects for the Evil Genius broadens your knowledge of this important, rapidly developing technology and shows you how to build practical, environmentally conscious projects using the three most popular and widely accessible fuel cells!
$24.95

Solar Energy Projects for the Evil Genius
by Gavin D J Harper



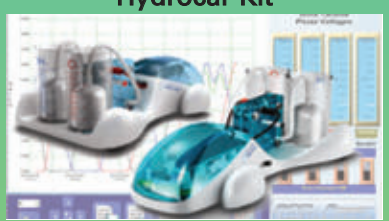
Let the sun shine on your evil side — and have a wicked amount of fun on your way to becoming a solar energy master! In *Solar Energy Projects for the Evil Genius*, high-tech guru Gavin Harper gives you everything you need to build more than 30 thrilling solar energy projects. You'll find complete, easy-to-follow plans, with clear diagrams and schematics, so you know exactly what's involved before you begin.
$24.95

B3-Phase Wind Turbine Kit

The Whirlybird Three-Phase Wind Turbine Kit is one of the great projects from the series of articles by John Gavlik, "Experimenting with Alternative Energy." In Parts 6 and 7, he teaches you how to produce the most electricity utilizing the wind. For kit details, please visit our webstore.

Subscriber's Price **$75.95**
Non-Subscriber's Price **$79.95**

Hydrocar Kit



The Hydrocar is used in a couple of great projects from the series of articles by John Gavlik, "Experimenting with Alternative Energy." In Parts 10 and 11, he teaches you the operation of the Polymer Electrolyte Membrane "reversible" fuel cell. For kit details and a demo video, please visit our webstore.

Subscriber's Price **$79.95**
Non-Subscriber's Price **$84.95**

Windpitch Kit

The WindPitch Wind Turbine Kit is a miniature real-working wind turbine and is one of the great projects from the series of articles by John Gavlik, "Experimenting with Alternative Energy." In Parts 8 and 9, he teaches you how to produce the most power by evaluating the pitch (setting angle) of the profiled blades. For kit details, please visit our webstore.

Subscriber's Price **$84.95**
Non-Subscriber's Price **$89.95**

FUNDAMENTALS FOR THE BEGINNER

A kit for these experiments can be purchased from the *Nuts & Volts* Webstore at www.nutsvolts.com or call us at 800-783-4624.

HOW AN SCR WORKS

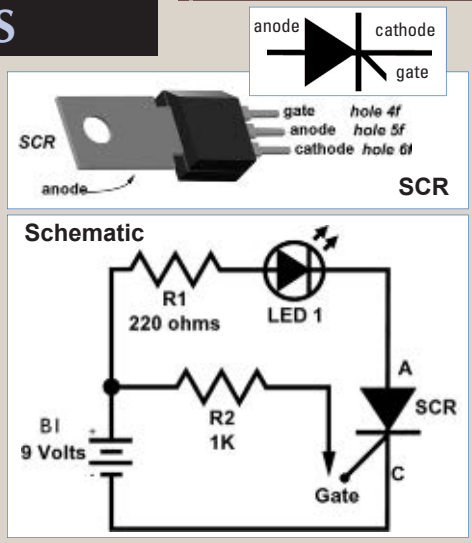
The function of a Silicon Controlled Rectifier (SCR) is to conduct electric current only after a positive pulse is put on its gate pin. The SCR will continue conducting until power is removed from the cathode and/or anode. SCRs are typically used in alarm system circuits.

1. Build the Circuit.

Using the schematic along with the pictorial diagram, place the components on a solderless breadboard as shown. Verify that your wiring is correct.

2. Do the Experiment.

Theory: In this simple circuit, the electrons will flow from the battery through the SCR, through the LED, and back to the battery *only* if the SCR receives a positive pulse on its gate. The positive pulse is put there through the 1,000 ohm resistor connected to the positive terminal on the nine-volt battery. (A 1,000 ohm resistor is used to make sure that not too much current flows to the gate.)



Procedure:

Connect a nine-volt battery to the battery snap. Touch the wire coming from the 1,000 ohm resistor to the gate of the SCR. This positive pulse from the battery will cause the SCR to conduct and light up the LED. It will remain lit even after you remove the pulse from the gate.

CLASSIFIEDS

KITS/PLANS

HIGH VOLTAGE KITS
Solid State Tesla Coils, High Voltage Kits, and Components
- Plasma Speakers
- Class-E Tesla Coils
- Audio Modulation Coils
- Lightning Detectors
- Flyback Drivers
- DRSSTCs
- Books

Plasma Speaker Kit
www.EasternVoltageResearch.com

High Quality Electronic Kits & Modules
$59.95
UK1133
50 AMP DIGITAL PWM
(OPTIONAL DISPLAY)
FM Transmitters
Timers
Audio Amplifiers
Motor Controllers
PIC Programmers / Arduino
• • • Custom Designs • • •
www.canakit.com

Electronic Kits
- Fun and Educational
- Discount White and Blue LEDs
- Quantity Discounts

www.bakatronics.com

QKITS.COM
Qkits Ltd
50 AMP PWM
1-888-GO 4 KITS

www.nutsvolts.com

NEW FOR 2010
VIDEO AGC/SYNC PROCESSING KIT
All Board Mounted Parts
w/ Instruction Manual CD
Choice of Non-programmed or Pre-programmed AVR Microcontroller
www.jlkelectronics.com

CELLULAR JAMMERS
Cellular Jammers and other unique plans and electronic kits
www.kenneke.com/nv

SOLAR PRODUCTS/KITS

BATTERY DESULFATORS
www.wizbangplus.com

ROBOTICS

MaxSonar
Ultrasonic Ranging is EZ

LV-MaxSonar-EZ
- Power-up calibration
- Smallest MaxSonar
- 1 inch resolution
- Low power, 2.5V-5.5V

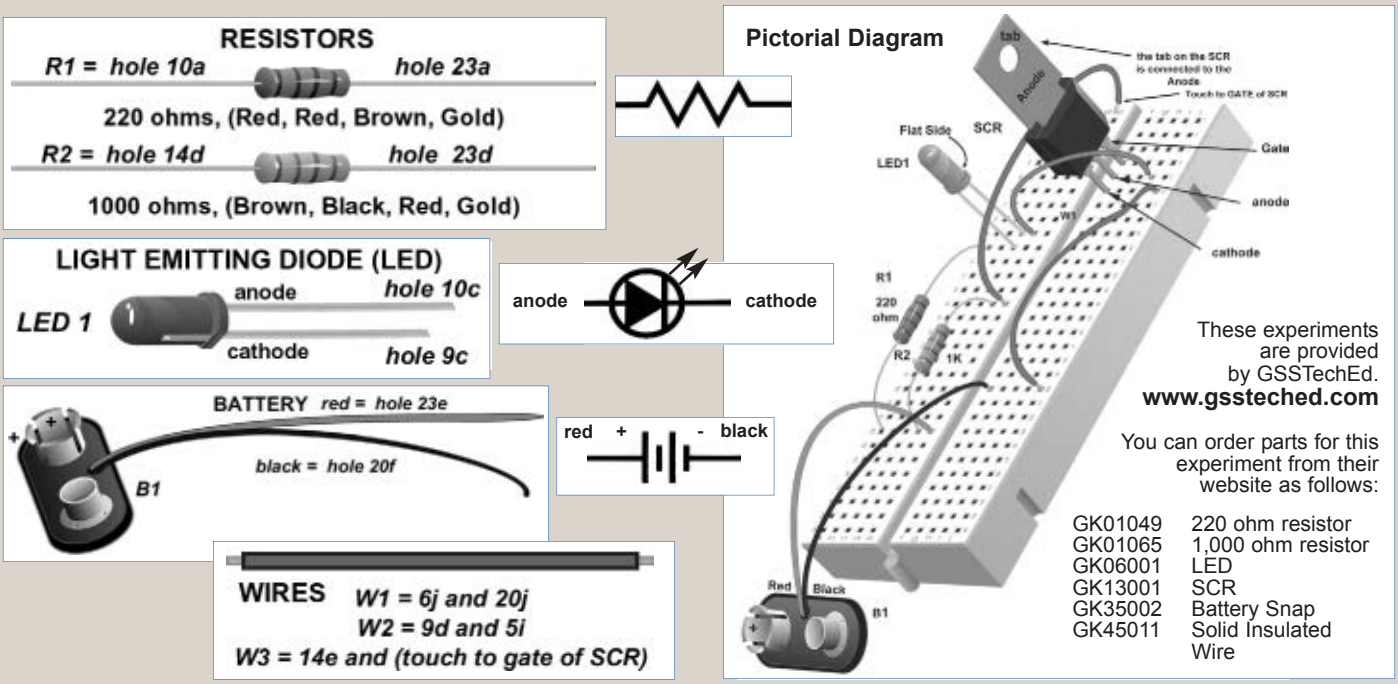
MaxSonar-WR (IP67)
- Industrial packaging
- Weather resistant
- Standard ¾" fitting
- Quality narrow beam

XL-MaxSonar-EZ
- Beam pattern choice
- Tiny size
- Real-time calibration
- High acoustic power

MaxSonar-WRC (IP67)
- Compact packaging
- Weather resistant
- Standard ¾" fitting
- Quality narrow beam

www.maxbotix.com

The New Evil
PICAXE MICROCONTROLLER PROJECTS FOR THE EVIL GENIUS
RON HACKETT
Available @ http://store.nutsvolts.com



COMPONENTS

RF PARTS™ CO.
Tubes, Transistors, Power Components
Email: rfp@rfparts.com • Web: www.rfparts.com
800-737-2787 Fax 888-744-1943

Brilldea
purveyor of prototyping goods
+RGB LED Ribbon
+RGB LED Pixels
+LED Painter
+Parallax Propeller
+LED Lumen
+3.0" LCD w/ composite video input
www.brilldea.com

uM-FPU V3.1
Floating Point Coprocessor

32-bit IEEE 754
SPI or I2C
DIP-18, SOIC-18
New full featured Compiler
Extensive floating point support, GPS input, FFT, 12-bit A/D, matrix operations, user-defined functions.
www.micromegacorp.com

Nixie Power

2-16V In, 4-8W / 150-200VDC Out
0.95" x 0.95" x 0.35"
$9.95 (Shipping included)
www.tayloredge.com/nixie

LEDs
Super Brights from 20¢
RGB Color Mixing 30 Pack $6.99
Assortment 120 Pack $24.99
Shipping from $3.00
Alan-Parekh.com/Store

MICRO CONTROLLERS

HARDWARE WANTED

DEC EQUIPMENT WANTED!!!
Digital Equipment Corp. and compatibles.
Buy - Sell - Trade
Call Keyways 937-847-2300
or email buyer@keyways.com

SURPLUS

SURPLUS ELECTRONIC PARTS & ACCESSORIES
Over 13,000 Items in Stock.
Cables, Connectors, Displays, Fans
Hardware, LEDs, Motors, Potentiometers
Relays, Semiconductors, Service Manuals, Speakers
Switches, Test Equipment, Tools, VCR Parts
Surplus Material Components
SMC ELECTRONICS
www.smcelectronics.com
No Minimum Order.
Credit Cards and PAYPAL Accepted.
Flat $4.95 per order USA Shipping.

Surplus - Obsolete - Used
electrical or electronic
find it here or sell it here !
no fees ~ no commissions ~ it's free !
www.SecondHandRadio.com

PCB SOFTWARE

EXCLUSIVE U.S. DISTRIBUTOR

YOUR COMPLETE SOLUTION!
• PCB Design
• PCB Simulation
• CAD/CAM Menu
121747, Hobby Vers., $175
121743, Pro. Version, $275
www.KELVIN.com

OSCILLOSCOPES

7 in 1 Scope !

CircuitGear CGR-101™ is a unique new, low-cost PC-based instrument which provides the features of seven devices in one USB-powered compact box: 2-ch 10-bit 20MS/sec 2MHz oscilloscope, 2-ch spectrum-analyzer, 3MHz 8-bit arbitrary-waveform/standard-function generator with 8 digital I/O lines. What's more – its open-source software runs with Windows, Linux and Mac OS's! **Only $199**
Saelig
1-888-7SAELIG
info@saelig.com
www.saelig.com

www.servomagazine.com

DESIGN/ENG SERVICES

*Circuit board layouts
*Prototype assemblies
WWW.OSPREYELECTRONICS.COM
Convert your sketch or print into a quality pcb for a reasonable price. Visit us on the web or call Osprey Electronics at (208) 664 1089 (USA)

worldmags

>>> QUESTIONS

Reactive Transformerless Power Supply

How does one calculate the value for the reactance capacitor in an AC to DC reactive power supply so as not to over-work the zener diode? I have used from .68 µf to 4.7 µf for various voltages out, but it has always been trial and error.

#10101 **Michael Stevens**
Rogersville, AL

Ping Tester

I work in large building complexes and need to "ping" devices several floors away or in the next building. I typically use a laptop, but it's difficult to hold or set down while in riser or mechanical rooms. There are cable testers with ping functionality but they are costly. It would be a neat project to build a ping tester using one of the widely available development boards and microcontrollers. There are so many choices I don't know where to start. Could someone point me in the right direction?

#10102 **Jim, via email**

Adapter Board

I recently got a Needham PB-10 EPROM programmer. It can program 4751H microcontrollers, but requires an adapter board to do so. I can't find any info on this board. Does anyone have a schematic or info so I could roll my own?

#10103 **Pierre Olivier, Quebec**

Resistance Inverter/Converter

I installed a dash from a Buick into a Winnebago. The gas gauge in the new dash requires 242 ohms for a full and 42 for an empty reading. The Winnebago sending unit gives a 10 ohms full and 180 ohms empty resistance reading. I need help in designing a circuit to do this conversion.

#10104 **Kenneth Lysek**
Henderson, CO

Racing Game Data Logging

I want to connect an analog tachometer and speedo or an instrument cluster to a PC via USB for receiving data from the Live For Speed racing game.

What Arduino and circuit components do I need to use on this project and which software is more useful?

#10105 **Ozgur**
Istanbul

>>> ANSWERS

[#4105 - April 2010] Steampunk Keyboard

*I am building a Steampunk keyboard for my own use. I would like to have a small motor spin some gears each time a key is pressed on the computer keyboard. I have not been able to locate an encoder output or an activity monitor that will show key presses. I don't care which keys are being pressed, and the gears spinning are just for show. I would like a hardware solution over a software solution – but a complete *.exe program would be acceptable. (Maybe flashing the scroll lock LED when keyboard activity is being transmitted to the keyboard controller.) Monitoring the data stream for F0 (Key up) occurrences would also work.*

The circuit diagram in **Figure 1** should do the trick; just connect 'IN' to the keyboard's data line. The PS/2 data line is normally held high, but is guaranteed to be pulled low at the beginning of a transfer. A simple monostable 555 can detect that low pulse and lengthen it to provide a control for your motor.

With the indicated values, the 555 should run the motor for 0.1s - 1.2s (depending on the setting of R2). I have included two options for connecting your motor. Option 1 should work for small motors that draw less than 0.2A and operate at 5V. If you have any other configuration, I recommend Option 2. If you feel that the time isn't long/short enough, just change the values of R1, R2, and C1 as needed (Max time = (R1 + R2) x C1 x 1.1; Min time = R1 x C1 x 1.1).

Daniel De Jager
Edgewood, WA

[#6101 - June 2010] Floppy Interface

I am looking to buy or get enough tips to design and build a device to do service adjustments on old style floppy drives that are used in music keyboards. To run analysis and adjustment software, I have to remove the drive from the keyboard and connect it to a computer. But, the new computers don't have floppy connectors any

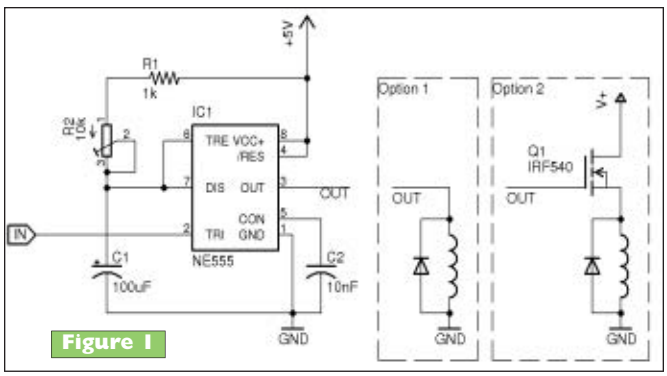


Figure 1

All questions *AND* answers are submitted by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving technical problems. Questions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted by readers and **NO GUARANTEES WHATSOEVER** are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals.

Always use common sense and good judgment!

worldmags

Send all questions and answers by email to **forum@nutsvolts.com**
*Check at **www.nutsvolts.com** for tips and info on submitting to the forum.*

more. The best device would have a floppy drive 34-pin connector on one end, a USB plug on the other, and some controller in the middle. The current USB external floppy drives are quite different from anything I could use for this project.

#1 The best way to approach this and probably the least problem-filled is to get an older PC — one that will run your adjustment software (lots of PIIIs running XP out there that you can buy for a song). Get a long 34-pin floppy cable, as well as a long power cable extension, open the PC up and connect the floppy cable to the motherboard, connect the power cable extension, pop one of the front covers off of the PC, snake the cables through the opening, then seal the PC back up. Most PC motherboards still make a provision for a floppy drive connection so that isn't a problem. A search on the Internet for the cables (try Cyberguys - **www.cyberguys.com** or Electronix - **www.electronix.com**) shouldn't be difficult. Now you have a test PC with the necessary cables hanging out the front of the unit. Unfortunately, they do not make an external floppy adapter like they do for making external IDE or SATA devices that use USB connectivity. What makes this setup nice is that if you can connect to the Internet with the PC, not only do you have a nice test jig you also have a PC to surf the Internet for your repair business.

Ralph J. Kurtz
Old Forge, PA

#2 You did not specify 3.5, 5.25, or 8" floppy drive. Here is something that should work:

The Buslink FDD1 3.5" USB floppy drive IS a standard 3.5" floppy drive with an easily accessible 34-pin ribbon cable in a case with a USB to floppy controller. Will it work with a 5.25" drive? I don't know yet, but it's the first time we have found a 3.5" USB floppy that isn't "integrated" and that consists of a USB to floppy controller with a standard 34-pin interface.

Several places sell these in the $40-$60 range. Try **www.pcrush.com/product/Floppy-Disk-Drives/590364/Buslink-FDD1-Floppy-Drive** or at Amazon Market place from **www1.shopping.com/xPO-Universal-Buslink-FDD-1**.

Barry Cole
Camas, WA

[#6102 - June 2010]
AC Motor Control

I'm building an alt-az antenna rotor controller from scratch, but I'm having trouble with the circuit for controlling the AC motor in the RadioShack rotator. I've tried using a TRIAC with little success. I believe they are my answer but I don't understand how to use them. What I need is a circuit that will accept one direction bit and one enable bit that will control a motor's direction of spin. The original controller acheived this by feeding one or the other non-neutral leads of the motor with ~22V AC; the third lead is common neutral.

#1 You may be having problems with ground referencing the gate signal or with the TRIAC eating up too much of the supply voltage (they do have a small voltage drop when they are ON).

One option that would fix both problems would be a pair of mechanical relays. They would exactly duplicate the pushbuttons in the rotor controller, so the rotor will be none the wiser and should work exactly as it always has. You would control the relay coils with a microprocessor and possibly a driver transistor. Don't forget to put diodes across the coils so the inductive kick doesn't hurt the driver or microprocessor output when the relay de-energizes.

If the problem is simply isolation or ground referencing, consider using opto-coupled TRIACs. They work just like the mechanical relay. You put their terminals across the wires that went to the pushbuttons. Their internal LED is the gate drive, so ground referencing is not a factor. They can also be driven from microprocessor outputs. Use a series resistor to limit the current to 20 mA or so. Either solution would be cheap — buy the parts for both and see what works best.

David Sarraf
Elizabethtown, PA

#2 Since you didn't supply information about the power requirements of the rotators, I just winged it. The schematic in **Figure 2** should be helpful.

Digi-Key sells a high power opto-coupler TRIAC for about $1.23 USD (751-1490-5-ND). If your rotators use less than 1A, this should do it for you.

Daniel De Jager
Edgewood, WA

[#6103 - June 2010]
Quartz Cookoo Clock

My quartz cookoo clock has a photodiode to turn off the clock at night, but I want it to work at night. Can I just remove the photodiode?

First of all, I suspect that the light sensor that turns off the cookoo sound at night is actually a CdS (cadmium sulfide) light sensor rather

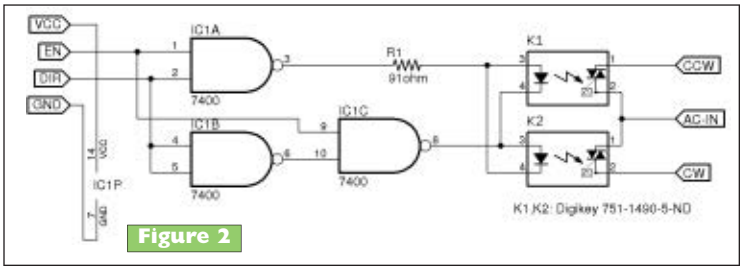


Figure 2

than a silicon photodiode as you stated. CdS sensors are generally used in consumer products since they are cost-effective and their relatively slow response time (compared to silicon) is acceptable. CdS sensors are light-dependent resistors and can be identified by a serpentine zig-zag sensor pattern. They have a high resistance (in the megohm region) in the dark and a much lower resistance (less than 10K) when illuminated. They are available in various sizes to provide a selection of sensitivity and dynamic range.

You can prevent your clock from going silent at night by simply bridging a resistor (try a value between 1K and 10K) across the CdS sensor in your clock. You could even put a switch in series with the resistor so that the clock could operate either normally or in a modified nighttime mode.

Bob Kovacs
Barnegat, NJ

[#6105 - June 2010]
Vacuum Cleaner Soft Start

I have a Hoover EmPower vacuum cleaner that causes the breakers in my home to trip almost every time I first power it on. After a few minutes of use, I can turn the machine off and back on with no problem. I am thinking that the initial inrush current is too much for the breakers (they are GFCI protected). Does anyone know of some sort of soft start circuit or any other trick that may help?

#1 The real problem is that the bearings in the vacuum cleaner motor are seizing, or that the lubrication is dried out. This makes the motor draw more (it's stalled) when starting cold. When the motor heats up, it spins more freely and draws less current. The motor needs repair. This may be as simple as taking it (the motor) apart, cleaning out the old dried oil and dirt, relubricating it, and reassembling. You might need to drill out some rivets to take it apart and replace with screws when you put it back together. I don't think a soft-start circuit will solve this problem.

Len S.
Los Altos, CA

#2 I doubt it's the inrush current, but a faulty GFCI. Older GFCIs were prone to tripping with motor loads and/or motors with brushes. I'd suggest replacing the GFCI breaker with a newer one. The specs do list 1,440 watts which does translate to 12A, so I would be attentive to what else is on the circuit.

A metal oxide resistor combined with a delay on make timer such as those made by **www.ssac.com** would work as a current limiter. The delay on make relay shorts the resistor out after a specified time delay. Unfortunately, the circuit would have to be at the vacuum cleaner. If the circuit worked, I'd almost be tempted to use a remote AC switch and place the device at the outlet. One example is the remote controls used for woodshop vacuums. I've used the technique before and had to experiment with resistor values and time.

Ron Dozier
Wilmington, DE

#3 There are at least two problems that can cause your circuit breakers to trip when using the vacuum. The first may be a worn out circuit breaker. As circuit breakers age, their springs weaken which causes them to trip on less than their rated current. Frequent tripping can accelerate the aging process. If that's the case, all you need to do is replace them with new breakers. Another possibility is a dusty motor. If you have GFCIs, they will trip on leakage current. Dust is hygroscopic which means it will absorb water vapor from the air. That slightly moist dust will leak current which can trip your GFCI. After the motor has run a bit, it warms up and drives off the moisture. The leakage current goes down and the GFCI won't trip. You didn't state what was tripping — the overcurrent part or the GFCI part. Either could be the culprit, and the fix should be fairly easy. I do not believe it is from too much current draw. The vacuum is rated for 12 amps which means that it should be okay on a standard 15 or 20 amp circuit. Motors do draw more on startup, but consider that this is a widely-used consumer product and a Google search shows nobody else having a problem with it.

David Sarraf
Elizabethtown, PA

#4 Ground Fault Interrupt (GFI) breakers are only installed in kitchen, bath, and outdoor outlets. Assuming that you are plugged into a bedroom or living room outlet, the GFI breaker is not the source of your problem. Do you have a relatively new house with arc fault breakers? The arc fault detecting breaker is known to sometimes trip on brushed motors such as used in vacuum cleaners. Try the vacuum cleaner on the outlet serving your refrigerator. It will not have an arc fault breaker because the reliable operation of the refrigerator trumps safety considerations. If the vacuum cleaner does not trip the refrigerator outlet, it is not the starting surge current; you need to plug into something other than an arc fault breaker.

Dennis Crunkilton
via email

[#6104 - June 2010]
Driveway Sensor

I want to set up a home driveway sensor similar to a traffic light actuator by using a single wire loop in the pavement that senses vehicles by changing the frequency of an oscillator. I need to know how it works and how to build it.

The unit works very much like a large metal detector, although the traffic signal units have more features.

I have access to many of these units if you have an interest. The traffic units have the ability to pulse (pulse the relay if a vehicle passes over or sits on the loop) a built-in relay, or they can place a constant signal to the relay (if a vehicle is sitting on the loop). Some have the ability to delay or extend the detected vehicle. Your loop will have to be about 4-6 turns of 14 Ga. wire in a 6' x 6' or smaller area. The detecter also has frequency and sensitivity, and will let you know of an out of range or faulty loop.

Ray
via email

worldmags

■ LOOK FOR ■ SEARCH FOR ■ FIND

Find your favorite advertisers here!

ADvertiser INDEX

AMATEUR RADIO AND TV

PolarisUSA Video, Inc. 13
Ramsey Electronics, Inc. 22-23
V-Module 30

BATTERIES/ CHARGERS

Cunard Associates 44

BUYING ELECTRONIC SURPLUS

Jaycar Electronics 31

CCD CAMERAS/ VIDEO

Circuit Specialists, Inc. 82-83
PolarisUSA Video, Inc. 13
Ramsey Electronics, Inc. 22-23

CIRCUIT BOARDS

AP Circuits 19
Cunard Associates 44
Dimension Engineering 35
Electronc Surplus 9
ExpressPCB 27
Front Panel Express LLC 37
PCB Pool 37
V-Module 30

COMPONENTS

Cana Kit Corp. 65
FTDI Chip 20
Fun Gizmos 44
Jameco 4
Linx Technologies 19
Mouser Electronics 21
superbrightlleds.com 36
V-Module 30

COMPUTER

Hardware

Electronc Surplus 9
FTDI Chip 20
Matrix Multimedia 44

Microcontrollers / I/O Boards

Abacom Technologies 57
Basic Micro 30
Fun Gizmos 44
Matrix Multimedia 44
Microchip 3
microEngineering Labs 12
Mouser Electronics 21
Net Media 2
Parallax, Inc. ....Back Cover
Pololu Robotics & Electronics 43
Solarbotics/HVW 8
Technological Arts 44
Trace Systems, Inc. 59

Software

Matrix Multimedia 44

DESIGN/ ENGINEERING/ REPAIR SERVICES

Cana Kit Corp. 65
ExpressPCB 27
Front Panel Express LLC 37
PCB Pool 37
Trace Systems, Inc. 59

EDUCATION

BaneBots 35
Command Productions 9
NKC Electronics 44
PAiA 37
Technological Arts 44

EMBEDDED TOOLS

Mouser Electronics 21
NetBurner 7

ENCLOSURES

Front Panel Express LLC 37
Integrated Ideas & Tech. 30

KITS & PLANS

Cana Kit Corp. 65
DesignNotes.com, Inc. 44
FTDI Chip 20
Jaycar Electronics 31
NetBurner 7
NKC Electronics 44
PAiA 37
QKITS 44
Ramsey Electronics, Inc. 22-23
Solarbotics/HVW 8

MISC./SURPLUS

All Electronics Corp. 45
Electronc Surplus 9
Front Panel Express LLC 37
Surplus Gizmos 12

MOTORS

BaneBots 35
Jameco 4

PROGRAMMERS

Basic Micro 30
microEngineering Labs 12

RF TRANSMITTERS/ RECEIVERS

Abacom Technologies 57
Linx Technologies 19

ROBOTICS

BaneBots 35
Fun Gizmos 44
Jameco 4
Lemos International Co., Inc. 44
Lynxmotion, Inc. 28
Net Media 2
Pololu Robotics & Electronics 43
Solarbotics/HVW 8

SATELLITE

Lemos International Co., Inc. 44

SECURITY

PolarisUSA Video, Inc. 13
Technological Arts 44

TEST EQUIPMENT

Circuit Specialists, Inc. 82-83
DesignNotes.com, Inc. 44
Dimension Engineering 35
Electronic Design Specialists 59
HAPRO Electronics 29
Jaycar Electronics 31
LeCroy 5
NKC Electronics 44
Trace Systems, Inc. 59

TOOLS

NetBurner 7

WIRE, CABLE AND CONNECTORS

DesignNotes.com, Inc. 44
Jameco 4

Abacom Technologies 57
All Electronics Corp. 45
AP Circuits 19
BaneBots 35
Basic Micro 30
Cana Kit Corp. 65
Circuit Specialists, Inc. 82-83
Command Productions 9
Cunard Associates 44
DesignNotes.com, Inc. 44
Dimension Engineering 35
Electronic Design Specialists 59
Electronc Surplus 9
ExpressPCB 27
Front Panel Express LLC 37
FTDI Chip 20
Fun Gizmos 44
HAPRO Electronics 29
Integrated Ideas & Tech. 30
Jameco 4
Jaycar Electronics 31
LeCroy 5
Lemos International Co., Inc. 44
Linx Technologies 19
Lynxmotion, Inc. 28
Matrix Multimedia 44
Microchip 3
microEngineering Labs 12
Mouser Electronics 21
NetBurner 7
Net Media 2
NKC Electronics 44
PAiA 37
Parallax, Inc. ....Back Cover
PCB Pool 37
PolarisUSA Video, Inc. 13
Pololu Robotics & Electronics 43
QKITS 44
Ramsey Electronics, Inc. 22-23
Solarbotics/HVW 8
superbrightlleds.com 36
Surplus Gizmos 12
Technological Arts 44
Trace Systems, Inc. 59
V-Module 30

worldmags

Triple Output Bench Power Supplies

The new CSI3303S & CSI5505S regulated DC power supplies are high reliability, variable DC Power Supplies with built in short circuit and thermal protection. These power supplies are suitable for the laboratory, electronics, communications equipment maintenance, production line, scientific research and educational institutions. Both units are equipped with protection circuits that protect the units from short circuits and over temperature by shutting the unit down for safety. Both units allow independent, serial and parallel mode operation.

Technical Specifications:

Independent mode: 2 independent 0-30V outputs

Series mode: CSI3303S Output from 0-60V & 0-3A
CSI5505S Output from 0-60V & 0-5A
Parallel mode: CSI3303S Output from 0-6A & 0-30V
CSI5505S Output from 0-10A & 0-30V

Both units also provide a 5V fixed output @ 3A

Load regulation: <0.1%+3mV (rating current <3A)
<0.2% +3mA
Ripple and noise: <1mVrms 5Hz-1MHz
<3mArms

Voltage accuracy: +/-0.5%rdg+2byte
Current accuracy: +/-.5%rdg+2byte

Display resolution: +/-0.5%rdg+2byte
Rated output: 5.0V +/-0.1V 3A

Tracking characteristics
Series specifications:
Load regulation: less than 50mV
Ripple and noise: (5Hz~1MHz) <=3mVRMS

Parallel characteristics:
Load regulation: less than 50mV
Ripple and noise: (5Hz-1MHz) CV less than 1mV<=6A), CV less than 1.5mV (I>6A)

Once again Circuit Specialists brings you a quality product at a great price!

Item# 0 to 3A **CSI3303S** $169.00
www.circuitspecialists.com/CSI3303S 4+ $155.00 ea.

Item# 0 to 5A **CSI5505S** $194.00
www.circuitspecialists.com/CSI5505S 4+ $179.00 ea.

Ideal for Law Enforcement, Post Fire Inspection, Plumbing, Facilities Maintenance, Security Companies, & Many Other Uses!

Aardvark II
Dual Camera
Wireless Inspection Camera
With Color 3.5" LCD Recordable Monitor
Your Extended Eyes & Hands!

See It!
Clearly in narrow spots, even in total darkness or underwater.

Find It!
Fast. No more struggling with a mirror & flash light.

Solve It!
Easily, speed up the solution with extended accessories.

Record It!
With the 3.5" LCD recordable monitor, you can capture pictures or record video for documentation.

Full specifications at
www.CircuitSpecialists.com/Aardvark

The Aardvark Wireless Inspection Camera is the only dual camera video borescope on the market today. With both a 17mm camera head that includes three atachable accessories and a 9mm camera head for tighter locations. Both cameras are mounted on 3ft flexible shafts. The flexible shaft makes the Aardvark great for inspecting hard to reach or confined areas like sink drains, AC Vents, engine compartments or anywhere space is limited. The Aardvark II comes with with a 3.5 inch color LCD monitor. The monitor is wireless and may be separated from the main unit for ease of operation. Still pictures or video can also be recorded and stored on a 2GB MicroSD card (included). The Aardvark's monitor also has connections for composite video output for a larger monitor/recorder and USB interface for computer connection. Also included is an AC adapter/charger, video cable and USB cable. Optional 3 ft flexible extensions are available to extend the Aardvark's reach (Up to 5 may be added for a total reach of 18 feet!).

Item # **AARDVARK** $249.00

3ft Extension AARDVARK-EXT $24.95

USB Digital Storage Oscilloscopes

* High performance:
* USB connected: Uses USB and supports plug'n play, with 12Mbp communication speed.
* Best performance for your dollar: Thease units have many features that are comparable to the high speedah stand-alone DSOs. But costs a fraction of the price.
* No external power required: Bus-powered from the host computers USB port.
* Probes & USB cable included.
* Easy to use: Intuitive and easy to understand.
* Various data formats: Can save wavrfrom in the following formats: .txt .jpg .bmp & MS excel/word

40MHz DSO-2090 www.circuitspecialists.com/DSO-2090 $169.00
60MHz DSO-2150 www.circuitspecialists.com/DSO-2150 $194.00
100MHz DSO-2250 www.circuitspecialists.com/DSO-2250 $249.00
200MHz DSO-5200 www.circuitspecialists.com/DSO-5200 $289.00
200MHz DSO-5200A www.circuitspecialists.com/DSO-5200A $355.00

Specifications	DSO-2090	DSO-2150	DSO-2250	DSO-5200 /5200A
Channels	2 Channels			
Impedence	1M 25pF			
Coupling	AC/DC/GND			
Vertical resolution	8 Bit			9 Bit
Gain Range	10mV-5V, 9 Steps			10mV-10V, 10Steps
DC Accuracy	+/- 3%			
Timebase Range	4ns - 1h 38 Steps			2ns-1h, 39 Steps
Vertical adjustable	Yes			
Input protection	Diode clamping			
X-Y	Yes			
Autoset	30Hz~40MHz	30Hz~60MHz	30Hz~100MHz	30Hz~200MHz
EXT. input	Yes			
Trigger Mode	Auto / Normal / Single			
Trigger Slope	+/-			
Trigger Level Adj.	Yes			
Trigger Type	Rising edge / Falling Edge			
Trigger Source	Ch1 / Ch2 / EXT			
Pre/Post trigger	0-100%			
Buffer size	10K-32K per ch			10K-512KB per ch
Shot Bandwidth	DC to 40MHz	DC to 60MHz	DC to 100MHz	100MHz
Max Sanple Rate	100MS/s	150MS/s	250MS/s	200MS/s / 250MS/s
Sampling Selection	Yes			
Waveform Display	port/line, waveform average, persistence, intensity			
Network	open / close			
Vertical Mode	Ch1, Ch2, Dual, Add			
CursorMeasurement	Yes			
Spectrum Analyzer				
Channels	2 Channels			
Math	FFT, addition, subtraction, multiplication, division.			
Bandwidth	40 MHz	60 MHz	100MHz	200 MHz
Cursor	Frequency, Voltage			
Data Samples	10K-32K/Ch			10K-1M/Ch

worldmags

We carry a LARGE selection of Power Supplies, Soldering Equipment, Test Equipment, Oscilloscopes, Digital Multimeters, Electronic Components, Metal and Plastic Project Boxes, Electronic Chemicals, PC Based Digital I/O Cards, Panel Meters, Breadboards, Device Programmers, and many other interesting items. Check out our website at: *www.CircuitSpecialists.com*

19inch Rack Mount Switch-Mode Power Supply with 3 Output Modes

NEW ITEM

The BTI-150 is a high efficency switching power supply. For use where multiple voltage continous power output is needed. Ideal for use in Technical Product Research, Laboratory, Educational Institutions, Telecommunication Industry, etc.

The Rack Mount Design makes it easy to stack multiple power supplies for high use areas.

Output Voltage	Output Current	V/C Display	Resolution
0~15V	0~10A	Digital	$0.1^{+/-}$2digit
0~30V	0~5A	Digital	$0.1^{+/-}$2digit
0~60V	0~2.5A	Digital	$0.1^{+/-}$2digit

Item # **BTI-150** **$189.00**

www.circuitspecialists.com/BTI-150

Programmable DC Electronic Loads

These devices can be used with supplies up to 360VDC and 30A. It features a rotary selection switch and a numeric keypad used to input the maximum voltage, current and power settings. These electronic DC loads are perfect for use in laboratory environments and schools, or for testing DC power supplies or high-capacity batteries. It also features memory, and can also be connected to a PC, to implement remote control and supervision.

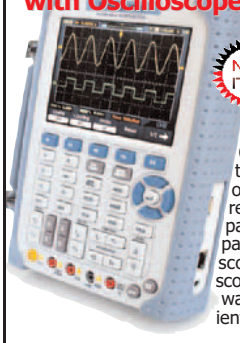
360V/150W (CSI3710A) $349.00

www.circuitspecialists.com/csi3710a

360V/300W (CSI3711A) $499.00

www.circuitspecialists.com/csi3711a

200MHz Hand Held Scopemeter with Oscilloscope & DMM Functions



NEW ITEM

Includes 1 Year USA Warranty

You get both a 200 MHz Oscilloscope and a multi function digital multimeter, all in one convenient lightweight rechargeable battery powered package. This power packed package comes complete with scopemeter, test leeds, two scope probes, charger, PC software, USB cable and a convenient nylon carrying case.

- 200MHz Handheld Digital Scopemeter with integrated Digital Multimeter Support
- 200MHz Bandwidth with 2 Channels
- 500MSa/s Real-Time Sampling Rate
- 50Gsa/s Equivalent-Time Sampling Rate
- 6,000-Count DMM resolution with AC/DC at 600V/800V, 10A
- Large 5.7 inch TFT Color LCD Display
- USB Host/Device 2.0 full-speed interface connectivity
- Multi Language Support
- Battery Power Operation (Installed)

Item # **DSO1200** **$739.00**

www.circuitspecialists.com/DSO1200

60MHz Hand Held Scopemeter with Oscilloscope & DMM Functions

- 60MHz Handheld Digital Scopemeter with integrated Digital Multimeter Support
- 60MHz Bandwidth with 2 Channels
- 150MSa/s Real-Time Sampling Rate
- 50Gsa/s Equivalent-Time Sampling Rate
- 6,000-Count DMM resolution with AC/DC at 600V/800V, 10A
- Large 5.7 inch TFT Color LCD Display
- USB Host/Device 2.0 full-speed interface connectivity
- Multi Language Support
- Battery Power Operation (Installed)

Item # **DSO1060** **$529.00**

www.circuitspecialists.com/DSO1200

60MHz Hand Held Scopemeter w/Oscilloscope, DMM Functions & 25 MHz Arbitrary Waveform Generator

- All the features of the DSO1060 plus a 25 MHz Arbitrary Waveform Generator.
- Waveforms can be saved in the following formats: jpg/bmp graphic file, MS excel/word file
- Can record and save 1000 waveforms
- DC to 25 MHz Arbitrary Waveform Generator

Item # **DSO-8060** **$659.00**

www.circuitspecialists.com/DSO-8060

Programmable DC Power Supplies

- Up to 10 settings stored in memory
- Optional RS-232, USB, RS-485 adapters
- May be used in series or parallel modes with additional supplies.
- Low output ripple & noise
- LCD display with backlight
- High resolution at 1mV

Model	CSI3644A	CSI3645A	CSI3646A
DC Voltage	0-18V	0-36V	0-72V
DC Current	5A	3A	1.5A
Power (max)	90W	108W	108W
Price	**$199.00**	**$199.00**	**$199.00**

BlackJack SolderWerks

Our Premium Line Up for Soldering, Repair & Rework

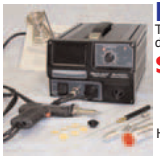
Rugged design at an affordable price..BlackJack SolderWerks from Circuit Specialists Inc. is the industry cost/performance leader and continues our reputation of providing high value products to our customers.

BK2000
Compact Soldering Station
$39.95

BK2000+
Compact Soldering Station with Digital Display
$54.95

BK2050+
70 Watt Soldering Station
$79.00

BK3000LF
Compact Lead Free Soldering Station
$59.95



BK4000
Thermostatically controlled desoldering station
$119.00

BK4050
HotAir System w/Vacuum IC handler & Mechanical Arm
$119.00

BK5000
Hot Air System w Soldering Iron & Mechanical Arm
$119.00

BK6000
Premium All-In-One Repairing Solder System
$229.00

www.circuitspecialists.com/blackjack

0-30V / 0-5A . DC Power Supply

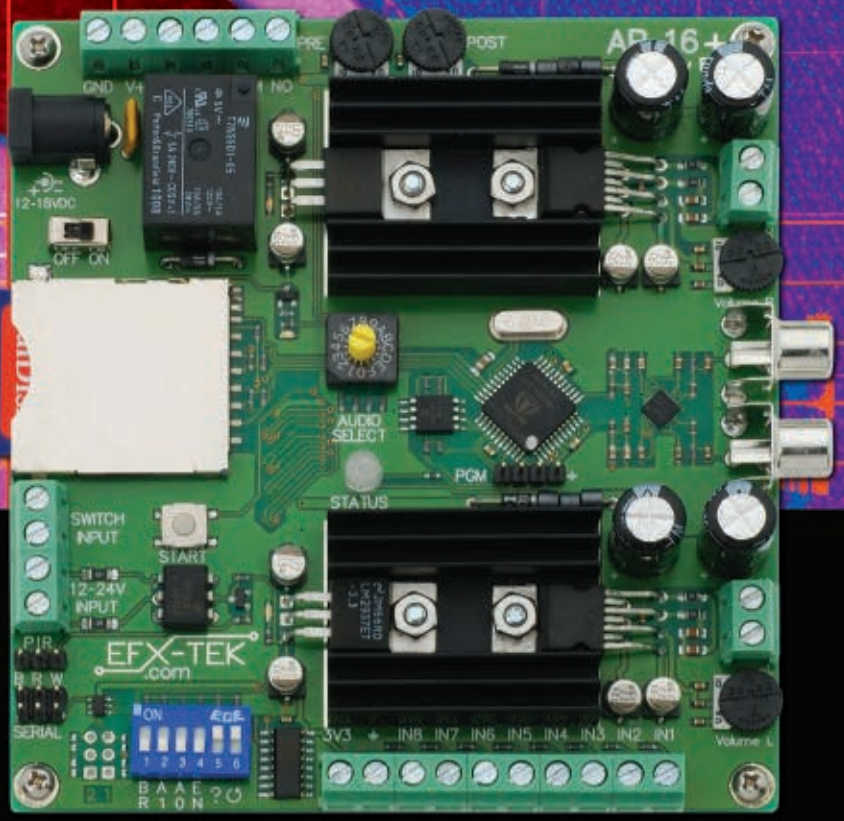
NEW ITEM

The CSI530S is a regulated DC power supply which you can adjust the current and the voltage continuously. An LED display is used to show the current and voltage values. The output terminals are safe 4mm banana jacks. This power supply can be used in electronic circuits such as operational amplifiers, digital logic circuits and so on. Users include researchers, technicians, teachers and electronics enthusiasts. A 3 ½ digit LED is used to display the voltage and current values.

ww.circuitspecialists.com/csi530s

Item # **CSI530S** **$84.95**

AP-16+ Audio Player



Now controlling powerful, booming sound is as simple as loading up your SD Card!

The Propeller-powered **AP-16+** (#31316; $129.99) by EFX-TEK allows you to play back high-quality, stereo WAV files for your standard SD card through two, booming 20 W amplifiers (one per channel). Advanced features like built-in randomization, auxiliary inputs, and the ability to use dry-contact, (optically isolated) 12 to 24 DC, or a PIR (or compatible) sensor put the AP-16+ in a class of its own. For stand-alone control the AP-16+ includes pre- and post-play delay potentiometers, as well as a relay that is active while your audio plays. Need background audio while your animated exhibit is idle? No problem, the AP-16+ includes an ambient loop control to do just that.

Order the Propeller-powered AP-16+ Audio Player online at www.parallax.com or call toll-free: 888-512-1024 (M-F, 7am-5pm, PDT).

Features:

- 8-core Propeller P8X32A-Q44 controller on board
- Plays stereo WAV files (16-bit, PCM, up to 44.1 kHz)
- Files stored on your standard SD card (FAT or FAT32 format)
- 2.1 mm, center-positive connector for DC power
- 24 stand-alone files under manual control
- Optional, auto-playing ambient file (AMBIENT.WAV)
- Manual input via (normally-open) dry-contact, (optically isolated) 12-24 VDC input, or PIR sensor
- Eight, dry-contact auxiliary inputs
- Relay output active when file plays (N.O. and N.C connections provided)
- Pre- and post-play delay potentiometers (manual mode)
- Dual 20 W amplifiers (one per channel)
- TTL serial control for connection to host microcontroller — unlimited files under serial control
- User-configured baud rate for serial control
- User-configured address allows up to four AP-16+ boards on one serial connection
- Easily updated as new features become available — update loads from an SD card (not included)
- Four 0.135 inch (3.4 mm) mounting holes

An 18 V 3.3 Amp Power Supply (sold separately; #750-00011; $34.99) is recommended for use with this unit.

PIR Sensor (#555-28027; $9.99) sold separately.

PARALLAX INC
www.parallax.com

"ParallaxInc" on Twitter, Facebook, and YouTube

Friendly microcontrollers, legendary resources.™

Prices subject to change without notice. Propeller, Parallax and the Parallax logo are trademarks of Parallax Inc. EFX-TEK is a trademark of EFX-TEK.

worldmags